AUTOMATIC DETERMINATION OF PART BUILD ORIENTATION FOR LASER POWDER BED FUSION ADDITIVE MANUFACTURING

YUCHU QIN

A thesis submitted to the University of Huddersfield in partial fulfilment of the requirements for the degree of Doctor of Philosophy

The University of Huddersfield

February 2021

Copyright statement

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns any copyright in it (the "Copyright") and he has given The University of Huddersfield the right to use such copyright for any administrative, promotional, educational and/or teaching purposes.

ii. Copies of this thesis, either in full or in extracts, may be made only in accordance with the regulations of the University Library. Details of these regulations may be obtained from the Librarian. This page must form part of any such copies made.

iii. The ownership of any patents, designs, trademarks and any and all other intellectual property rights except for the Copyright (the "Intellectual Property Rights") and any reproductions of copyright works, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property Rights and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property Rights and/or Reproductions.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my main supervisor, Professor Paul J. Scott, who has provided me the precious opportunity to engage in the study of computer-aided additive manufacturing with sufficient financial support. His profound knowledge in mathematics and computational geometry and rich experience in solving real-world manufacturing problems, are of tremendous help for my research work and thesis.

I would also like to present my hearty gratitude to my co-supervisors, Professor Dame Xiangqian (Jane) Jiang and Dr. Qunfen Qi, for providing me valuable guidance, selfless help and adequate financial support. Their extensive knowledge in advanced manufacturing and metrology and rich experience in academic research, are an important contribution to my research work and thesis.

Special thanks go to Professor Meifa Huang and Professor Yanru Zhong from Guilin University of Electronic Technology, who led me to the door of scientific research on mechanical engineering and developed a forward-looking plan for my research career ten years ago.

My thanks go to Professor Zhijie Xu, Dr. Wenhan Zeng and Dr. Andrew Townsend, who made constructive comments on my research work. My thanks also go to Dr. Peizhi Shi, who provided instruction and help in implementing the proposed methods, Dr. Luca Pagani, who provided instruction in plotting figures in Matlab, and Dr. Yicha Zhang from the University of Technology of Belfort-Montbéliard, who provided the STL files of two 3D models.

I would also like to thank Professor Liam Blunt, Dr. Feng Gao, Professor Zonghua Zhang, Professor Tong Guo, Dr. Tukun Li, Dr. Shan Lou, Dr. Andrew Henning, Dr. Hussam Muhamedsalih, Professor Guoqin Huang, Dr. Tianqi Gu, Dr. Dawei Tang, Dr. Zhen Tong, Dr. Hongyu Ren, Dr. Wenbin Zhong, Dr. Yongjia Xu, Dr. Tao Zhang, Dr. Zhoulong Li, Dr. Duo Li, Ms. Yue Liu, Ms. Zhenting Zhang, Mr. Shubhavardhan Narasimharaju and all other members and visiting scholars of the Centre for Precision Technologies, who created an enjoyable academic atmosphere.

Last, but certainly not least, I own my sincere gratitude to my parents for their upbringing, and my beloved wife Dr. Xiaolan Cui for her understanding and support, with whom I can share my joy of success and from whom I can gain motivation and encouragement.

List of publications and statement of reproduction

List of refereed journal papers:

1. **Qin, Y.**, Qi, Q., Shi, P., Scott, P.J. and Jiang, X., 2021. Status, issues, and future of computer-aided part orientation for additive manufacturing. The International Journal of Advanced Manufacturing Technology. Under Review.
2. **Qin, Y.**, Qi, Q., Scott, P.J. and Jiang, X., 2019. Status, comparison, and future of the representations of additive manufacturing data. Computer-Aided Design, 111(6), 44−64.
3. **Qin, Y.**, Qi, Q., Shi, P., Scott, P.J. and Jiang, X., 2020. Automatic generation of alternative build orientations for laser powder bed fusion based on facet clustering. Virtual and Physical Prototyping, 15(3), 307−324.
4. **Qin, Y.**, Qi, Q., Scott, P.J. and Jiang, X., 2019. Determination of optimal build orientation for additive manufacturing using Muirhead mean and prioritised average operators. Journal of Intelligent Manufacturing, 30(8), 3015−3034.
5. **Qin, Y.**, Qi, Q., Shi, P., Scott, P.J. and Jiang, X., 2021. Automatic determination of part build orientation for laser powder bed fusion. Virtual and Physical Prototyping, 16(1), 29−49.
6. **Qin, Y.**, Qi, Q., Scott, P.J. and Jiang, X., 2019. Multi-criteria group decision making based on Archimedean power partitioned Muirhead mean operators of q-rung orthopair fuzzy numbers. Plos One, 14(9), e0221759.
7. **Qin, Y.**, Qi, Q., Scott, P.J. and Jiang, X., 2020. Multiple criteria decision making based on weighted Archimedean power partitioned Bonferroni aggregation operators of generalised orthopair membership grades. Soft Computing, 24(16), 12329−12355.
8. **Qin, Y.**, Qi, Q., Scott, P.J. and Jiang, X., 2020. An additive manufacturing process selection approach based on fuzzy Archimedean weighted power Bonferroni aggregation operators. Robotics and Computer-Integrated Manufacturing, 64(8), 101926.
9. **Qin, Y.**, Qi, Q., Shi, P., Scott, P.J. and Jiang, X., 2020. Novel operational laws and power Muirhead mean operators of picture fuzzy values in the framework of Dempster-Shafer theory for multiple criteria decision making. Computers & Industrial Engineering, 149(11), 106-853.
10. **Qin, Y.**, Qi, Q., Shi, P., Scott, P.J. and Jiang, X., 2020. Linguistic interval-valued intuitionistic fuzzy Archimedean prioritised aggregation operators for multi-criteria decision making. Journal of Intelligent & Fuzzy Systems, 38(4), 4643−4666.

11. Shi, P., Qi, Q., **Qin, Y.**, Scott, P.J. and Jiang, X., 2021. Intersecting Machining Feature Localization and Recognition via Single Shot Multibox Detector. IEEE Transactions on Industrial Informatics, 17(5), 3292−3302.
12. Shi, P., Qi, Q., **Qin, Y.**, Scott, P.J. and Jiang, X., 2020. A novel learning-based feature recognition method using multiple sectional view representation. Journal of Intelligent Manufacturing, 31(5), 1291−1309.
13. Zhong, Y., Jiang, C., **Qin, Y.**, Yang, G., Huang, M. and Luo, X., 2019. Automatically generating assembly sequences with an ontology-based approach. Assembly Automation, 40(2), 319−334.
14. Zhong, Y., Cao, L., Zhang, H., **Qin, Y.**, Huang, M. and Luo, X., 2021. Hesitant fuzzy power Maclaurin symmetric mean operators in the framework of Dempster–Shafer theory for multiple criteria decision making. Journal of Ambient Intelligence and Humanized Computing, https://doi.org/10.1007/s12652-021-02932-4.
15. Zhong, Y., Guo, X., Gao, H., **Qin, Y.**, Huang, M. and Luo, X., 2019. A new multi-criteria decision-making method based on Pythagorean hesitant fuzzy Archimedean Muirhead mean operators. Journal of Intelligent & Fuzzy Systems, 37(4), 5551−5571.
16. Zhong, Y., Gao, H., Guo, X., **Qin, Y.**, Huang, M. and Luo, X., 2019. Dombi power partitioned Heronian mean operators of q-rung orthopair fuzzy numbers for multiple attribute group decision making. Plos One, 14(10), e0222007.
17. Tang, Z., Huang, M., Sun, Y., Zhong, Y. and **Qin, Y.**, 2019. Rapid evaluation of coaxiality of shaft parts based on double maximum material requirements. Measurement, 147, 106868.
18. Tang, Z., Huang, M., Sun, Y., Zhong, Y., **Qin, Y.** and Huang, J., 2019. Coaxiality evaluation based on double material condition. Measurement, 141, 287−295.

Statement for reproduction of publications:

The main content of Paper 1 to Paper 8 in the list above has been reproduced in this thesis:

- The literature review in Paper 1 to Paper 5 was reproduced in Chapter 1 and Chapter 2;
- The main content of Paper 3 was reproduced in Chapter 3;
- The main content of Paper 6 to Paper 8 was reproduced in Chapter 4;
- The main content of Paper 4 and a part of Paper 5 were reproduced in Chapter 5.
- The main content of Paper 5 was reproduced in Chapter 6 and Chapter 7.

For each of the eight papers, I am the lead author and was the sole PhD student. I carried out all research work and wrote and revised the manuscript with minor changes from my co-authors.

Abstract

Laser powder bed fusion is one of the key and most widely used additive manufacturing processes. The use of this process to build a part includes a set of continuous activities, where process planning is an indispensable one. This activity refers to a systematic planning of the build orientation, supports, slices, laser scanning path and process parameters to build a part using a laser powder bed fusion machine. It includes four successive steps, where build orientation determination is the first step. At present, most of the determination tasks in real workshops are manually completed by process planners according to their production knowledge and experience. Different process planners could determine different build orientations for an identical part under the same conditions. This would increase the build time and build cost and have a negative influence on the quality and production stability of the built part.

To this end, a study on automatic determination of part build orientation for laser powder bed fusion additive manufacturing is carried out in this thesis. This study divides build orientation determination into alternative orientation generation and optimal orientation selection. Firstly, an automatic alternative orientation generation method based on facet clustering for laser powder bed fusion is presented. A set of fuzzy aggregation operators for evaluating the values of attributes of alternative orientations are then constructed. Using the constructed operators, an automatic optimal orientation selection method based on multi-attribute decision making for laser powder bed fusion is proposed. Finally, an automatic part build orientation determination method for laser powder bed fusion is developed via combining and implementing the alternative orientation generation method and optimal orientation selection method. Case studies are presented to illustrate the application of the developed method. The effectiveness, efficiency and advantages of the method are evaluated via theoretical analysis, experimental analysis and comparisons.

The completed research work in the thesis is expected to realise a transformation of part orientation for laser powder bed fusion from a manual mode to a computer-aided mode. It can easily be extended to other additive manufacturing processes and can provide effective ideas and methodology for study of computer-aided process planning for additive manufacturing.

Table of content

List of acronyms

AA	Arithmetic Average
ABO	Alternative Build Orientation
AG	Additive Generator
AM	Additive Manufacturing
AMF	Additive Manufacturing File
ASCII	American Standard Code for Information Interchange
ASTM	American Society for Testing Material
ATT	Algebraic T-norm and T-conorm
AO	Aggregation Operator
BM	Bonferroni Mean
CAD	Computer-Aided Design
CI	Consistency Index
CLI	Common Layer Interface
CR	Consistency Ratio
ETT	Einstein T-norm and T-conorm
FTT	Frank T-norm and T-conorm
GA	Geometric Average
GBM	Geometric Bonferroni Mean
GMM	Geometric Muirhead Mean
GMSM	Geometric Maclaurin Symmetric Mean
HDBSCAN	Hierarchical Density-Based Spatial Clustering of Applications with Noise
HPGL	Hewlett-Packard Graphics Language
HTT	Hamacher T-norm and T-conorm
ISO	International Organisation for Standardisation
LPBF	Laser Powder Bed Fusion
MADM	Multi-Attribute Decision Making
MM	Muirhead Mean
MOO	Multi-Objective Optimisation
MSM	Maclaurin Symmetric Mean
OBJ	Wavefront Object

OBO	Optimal Build Orientation
PA	Power Average
RI	Random Index
SLC	Stereo Lithography Contour
STL	Stereolithography or Standard Tessellation Language
2D	Two-Dimensional
3D	Three-Dimensional
3MF	3D Manufacturing Format
2.5D	Two-Point-Five-Dimensional
FAWPAA	Fuzzy Archimedean Weighted Power Arithmetic Average
FAWPBM	Fuzzy Archimedean Weighted Power Bonferroni Mean
FAWPGA	Fuzzy Archimedean Weighted Power Geometric Average
FAWPGBM	Fuzzy Archimedean Weighted Power Geometric Bonferroni Mean
FAWPGMM	Fuzzy Archimedean Weighted Power Geometric Muirhead Mean
FAWPGMSM	Fuzzy Archimedean Weighted Power Geometric Maclaurin Symmetric Mean
FAWPMM	Fuzzy Archimedean Weighted Power Muirhead Mean
FAWPMSM	Fuzzy Archimedean Weighted Power Maclaurin Symmetric Mean

List of figures

List of tables

1 Introduction

1.1 Background

Additive manufacturing (AM), which in the past was called as additive fabrication, additive processes, additive techniques, additive layer manufacturing, layer manufacturing, solid freeform fabrication, freeform fabrication or three-dimensional (3D) printing, refers to the processes used to manufacture 3D objects, in which materials are accumulated layer by layer via specific techniques such as extrusion, sintering, melting, photopolymerisation, jetting, lamination and deposition (ISO/ASTM 52900, 2015). Modern AM processes first emerged with stereolithography in the 1980s and have been used for prototype production since then (Gibson et al., 2015). Recently, the development of computer-aided design (CAD), material unit control, material processing and forming, equipment recoating and efficient manufacturing technologies made AM processes applicable to fabricate end-use products. AM processes have a set of advantages such as providing high design flexibility, achieving geometric complexity without additional cost, generating fewer waste material and avoiding non-essential assembly over traditional subtractive manufacturing technologies (Bourell et al., 2009; Attaran, 2017). Further, AM processes make it possible to manufacture products with heterogeneous materials and customisable functions. Convinced by such advantages and potential, some have anticipated that AM processes would bring revolutionary changes to the manufacturing industry (Gao et al., 2015).

Existing AM processes were divided into seven categories, where powder bed fusion is one of them (ISO 17296-2, 2015). In this process, either laser beam, electron beam or heat is used to fuse the material together to construct a 3D object. According to the energy source, powder bed fusion can be further classified into selective laser sintering, selective heat sintering, direct metal laser melting, selective laser melting and electron beam melting, where direct metal laser melting and selective laser melting are collectively referred to as laser powder bed fusion (LPBF) (King et al., 2014; Gibson et al., 2015). LPBF is one of the key AM processes which can be applied to produce fully dense products with high accuracy, strength and stiffness. This process is suitable for high-value industries such as aerospace, defence, automotive industry and medical prosthetics (King et al., 2015; Yap et al., 2015).

The application of the LPBF process to build a part includes a set of continuous activities, where process planning is an indispensable one (Kim et al., 2015). Process planning for LPBF AM is the planning of the build orientation, supports, slices, laser scanning path and process parameters to build a part using an LPBF machine on the basis of the 3D model data of the part

(Kulkarni et al., 2000; Ahsan et al., 2015). This activity consists of four successive preparation steps before building the part, in which build orientation determination is the first step. This step has a direct influence on its three subsequent steps, namely support generation, 3D model slicing and path planning (Liang, 2018).

In the context of LPBF AM, the build orientation of a part is a critical process variable for building the part, because it directly affects the time and cost to build the part and the property, accuracy and surface quality of the built part (Edwards and Ramulu, 2014; Snyder et al., 2015; Wauthle et al., 2015; Calignano, 2018; Zhou and Ning, 2020; Di Angelo et al., 2020). Build orientation determination for LPBF AM is the determination of a desirable build orientation of an LPBF part according to the 3D model of the part and specific production requirements and preferences on the part. It can be addressed by using a two-step approach. The first step is to generate a certain number of alternative build orientations (ABOs) from an infinite number of possible orientations. The second step is to select the optimal build orientation (OBO) from the generated ABOs (Kulkarni et al., 2000). In real workshops, most build orientation determination tasks for LPBF AM are manually completed by process planners according to their production knowledge and experience. Different process planners could possibly determine different build orientations for an identical part under the same conditions. This would increase the build time and build cost and have a negative effect on the quality and production stability of the built part.

To this end, a study on automatic determination of part build orientation for LPBF AM is carried out in this thesis. An accelerated HDBSCAN* (hierarchical density-based spatial clustering of applications with noise*) algorithm (Campello et al., 2015; McInnes and Healy, 2017) is introduced to develop an automatic ABO generation method, and a multi-attribute decision making (MADM) method based on fuzzy aggregation operators (AOs) (Greco et al., 2016; Mardani et al., 2018) is presented to develop an automatic OBO selection method. An automatic determination method is obtained by combining the developed methods. It is envisaged that the automatic determination method will free LPBF AM process planners from manual determination of part build orientation to automatic determination aided by computers, and will provide effective ideas and methodology for research and implementation of computer-aided process planning for AM within the academia and the industry.

1.2 Aim and objectives

The aim of the thesis is to develop a methodology for automatic generation and evaluation of ABOs to determine the OBO in LPBF AM. The objectives of the thesis are outlined as follows:

- To provide a brief overview of AM, AM processes, LPBF and LPBF product realisation process activities and a comprehensive review of the existing research work on build orientation determination for AM.
- To present an automatic ABO generation method based on facet clustering for LPBF AM. This involves clustering planar triangular facets, generating the ABOs of each facet cluster and combining the ABOs of all facet clusters.
- To construct a set of fuzzy AOs for comprehensively evaluating the values of attributes of ABOs. This involves establishing a general expression for each AO, exploring the mathematical property of each AO and deriving the specific expressions for each AO.
- To propose an automatic OBO selection method based on fuzzy MADM for LPBF AM. This would involve identifying the attributes of ABOs, estimating and fuzzifying of the values of attributes, determining the relationships among attributes and the weights of attributes, generating a ranking of ABOs and selecting the OBO.
- To develop an automatic build orientation determination method for LPBF AM and demonstrate its application, effectiveness, efficiency and advantages.

1.3 Thesis structure

The remainder of the thesis is organised as follows:

- Chapter 2 provides a brief overview of AM, AM processes, LPBF and LPBF product realisation process activities and a comprehensive review of the existing build orientation determination methods for AM with an analysis of the research gaps.
- Chapter 3 describes the details of an automatic ABO generation method based on facet clustering for LPBF AM.
- Chapter 4 states the details of a set of fuzzy AOs for comprehensively evaluating the values of attributes of ABOs.
- Chapter 5 explains the details of an automatic OBO selection method based on fuzzy MADM for LPBF AM.
- Chapter 6 documents the development of an automatic build orientation determination method for LPBF AM and the demonstration of this method.
- Chapter 7 summarises the research work and major contributions of the thesis and suggests some future research directions based on the limitations of the developed methodology.

2 Literature review

The research topics involved in build orientation determination for LPBF AM are AM, AM processes, LPBF, LPBF product realisation process and build orientation determination. In this chapter, a brief overview of AM, categorised AM processes, LPBF and LPBF product realisation process activities is first provided. Then a comprehensive review of the existing build orientation determination methods for AM is presented. Based on the review, the research gaps of the thesis are identified.

2.1 Additive manufacturing (AM)

In this section, the definition of the term AM is first introduced. Then the history of AM is reviewed. Finally, the distinguishing characteristics of AM are noted.

2.1.1 Definition of AM

While the terms freeform fabrication, solid freeform fabrication, layer manufacturing, additive layer manufacturing, additive processes, additive techniques, 3D printing and additive manufacturing all describe a similar set of processes, their usage has evolved with the field (Weber et al., 2013). Today, 3D printing and additive manufacturing are two prevailing terms. According to the latest international standard of AM related terms ISO/ASTM 52900 (2015), the two terms are respectively defined as follows:

- 3D printing: “Fabrication of objects through the deposition of a material using a print head, nozzle or another printer technology”.
- Additive manufacturing: “Process of joining materials to make objects from 3D model data, usually layer upon layer, as opposed to subtractive manufacturing and formative manufacturing methodologies”.

It is difficult to distinguish the two terms from such definitions. The standard appends a note after the definition of 3D printing. The note shows that the term 3D printing is generally used in a non-technical context synonymously with the term additive manufacturing. But until recently the term 3D printing has in particular been associated with machines which are low end in price or overall capability. As most relevant international standards tend to use the term additive manufacturing, the present thesis also uses this term. In the thesis, additive manufacturing, or AM, refers to the processes used to manufacture 3D objects, in which materials are accumulated layer upon layer through certain techniques such as extrusion, sintering, melting, photopolymerisation, jetting, lamination and deposition.

2.1.2 History of AM

With the advent of photography in 1800s, scientists began to contemplate how to extend the technique to replicate 3D physical objects. Contemporary additive techniques were photo sculpture (in the 1860s) and topography (in the 1890s). These early techniques led to the invention of photo-glyph recording technology (patented in 1956), which exposed layers of a transparent photo emulsion selectively when scanning cross-sections of the object to be replicated (Weber et al., 2013). In 1981, Hideo Kodama in Nagoya Municipal Industrial Research Institute invented two additive techniques to fabricate 3D plastic models using photo-hardening thermoset polymer, in which the ultraviolet exposure field was under the control of a scanning fibre transmitter. On July 16, 1984, Alain Le Mehaute, Olivier de Witte and Jean Claude Andre applied their patent for a stereolithography process. Three weeks later, Charles Hull in 3D Systems Corporation filed his patent for a stereolithography system, in which layers were stacked via selectively curing photopolymer with ultraviolet light and a 3D object was generated by creating a cross-sectional pattern of the object. The major contributions of this patent were the STL (Stereolithography or Standard Tessellation Language) file format (Roscoe, 1988) and the slicing and infill strategies, which are still used in almost all AM processes today.

Since the development of the stereolithography system in 1984, many processes that pave the way for what is today known as AM have been developed. Some representative ones are chronologically listed as follows:

- Selective laser sintering. In 1986, Carl Deckard and Joe Beaman at the University of Texas at Austin developed a selective laser sintering process to build 3D objects using a laser beam as the energy source to sinter powder material.
- Laminated object manufacturing. In 1987, Michael Feygin and his team in Helisys (now Cubic Technologies) patented a laminated object manufacturing technique which forms integral objects from sheet laminations.
- Fused deposition modelling. In 1988, Scott Crump in Stratasys Ltd developed a fused deposition modelling process which builds layers via mechanically extruding molten thermoplastic material onto a substrate.
- 3D printing. In 1989, Emanuel Sachs and his team at the Massachusetts Institute of Technology developed a 3D printing technology which injects the binding agent and coloured ink on a powder bed using the injectors of a traditional ink-jet printer.
- Binder jetting. In 1993, Ely Sachs and Mike Cima at the Massachusetts Institute of Technol-

ogy developed a binder jetting process in which 3D objects are fabricated by selectively depositing liquid binder.

- Electron beam melting. In 1993, Arcam AB patented an electron beam melting technology in which the material is placed in a vacuum and melted and fused together by an electron beam.
- Direct metal laser sintering (melting). In 1994, EOS developed a direct metal laser sintering process that leverages a laser beam to melt and fuse metallic powders together. The name of this process is misleading, as the powder is being melted when building a part. The name direct metal laser melting is more appropriate for the process.
- Selective laser melting. In 1995, the Fraunhofer Institute introduced a selective laser melting process whose working principle is very similar to direct metal laser melting.
- Laser engineered net shaping. In 1995, the Sandia National Laboratories developed a laser engineered net shaping technique in which 3D objects are built via using focused thermal energy to melt material as it is being deposited.
- Material jetting. In 1999, the Objet Geometries Ltd (merged with Stratasys in 2012) patented a material jetting process in which 3D objects are built via selectively depositing the droplets of material.
- Paper 3D printing. In 2003, Conor MacCormack and Fintan MacCormack at MCor Technologies developed a paper 3D printing technique which is based on sheet lamination.
- Contour crafting. In 2010, Behrokh Khoshnevis at the University of Southern California patented a contour crafting technique which uses a computer-controlled crane or gantry to construct edifices rapidly and efficiently.
- Selective heat sintering. In 2011, Blueprinter launched a selective heat sintering technology at Euromold which uses thermal energy to replace laser beam in selective laser sintering.
- 3D concrete printing. In 2014, Richard Buswell and his team at the Loughborough University succeeded in licensing a 3D concrete printing technology to Skanska which can manufacture full-scale construction and architectural components.

In 2015, the International Organisation for Standardisation (ISO) and the American Society for Testing Material (ASTM) officially defined the set of processes which join materials layer upon layer to make 3D objects from 3D model data as AM in their issued standard ISO/ASTM 52900 (2015). Nowadays, the volume of AM technologies is so huge that it is very difficult to provide an exhaustive review. For a more detailed history of AM, please refer to the reports of Weber et al. (2013) and Wohlers and Gornet (2016).

2.1.3 Characteristics of AM

Distinguishing characteristics of AM processes are usually presented via a comparison with conventional subtractive manufacturing technologies. On the basis of the surveys of Bourell et al. (2009), Gao et al. (2015) and Attaran (2017), the unique capabilities of AM processes are noted as follows:

- Provide high design flexibility. A common characteristic of AM processes is their layer upon layer building manner, which theoretically enables the creation of arbitrary geometric shapes. This is different from subtractive processes, which limit the freedom of design because of the need for various fixtures and machining tools and the difficulty of the cutter in reaching deep and invisible areas.
- Achieve geometric complexity without additional cost. For AM processes, the complexity of geometric shapes comes at no additional cost, as they do not require additional tooling or re-fixturing. In subtractive manufacturing processes, geometric complexity can be achieved via injection molding. However, the more complex the geometric shapes, the higher the cost of injection molding.
- Generate fewer waste material. On average, manufacturing parts using AM processes generates considerably less waste material than using subtractive manufacturing processes.
- Avoid non-essential assembly. AM technologies enable production of products which would need assembly of a set of components if manufactured conventionally.
- Produce products with multiple materials and customisable functions. An intriguing characteristic of AM processes is that they enable the direct production of products with heterogeneous materials, multiple colours and customisable functions, while it is impossible for subtractive manufacturing processes.

2.2 Categorised AM processes

In 1991, Kruth (1991) classified AM processes into liquid-based process, powder-based process and solid-based process from the perspective of fabrication material. A complete family tree of AM processes was constructed in German manufacturing process standard DIN 8580 in 2003. After that, Williams et al. (2011) presented a functional classification schema of AM techniques. Most recently, ISO has categorised AM processes into vat photopolymerisation, material jetting, binder jetting, powder bed fusion, material extrusion, directed energy deposition and sheet lamination (ISO 17296-2, 2015). In this section, a brief introduction of the development and characteristics of each of the seven processes is provided.

2.2.1 Vat photopolymerisation

Vat photopolymerisation is an AM process in which 3D objects are built by selectively curing the liquid photopolymer in a vat through targeted light-activated polymerisation (ISO/ASTM 52900, 2015). The earliest vat photopolymerisation technology is stereolithography, which was discovered by Japanese and French researchers in the early 1980s and used to develop the first commercial AM machine by Charles Hull in 3D Systems Corporation in 1984. Since the advent of stereolithography, vat photopolymerisation has grown to include digital light processing, continuous liquid interface production and daylight polymer printing. So far, over 440 industrial AM machines using vat photopolymerisation have been developed (Senvol LLC, 2020).

Vat photopolymerisation is known as a fast and accurate AM process that can be leveraged to build rather large objects. However, this process is inherently limited to polymer because of its dependence on photopolymerisation. By suspending nanoparticles in the liquid photocurable resin, the process has been extended to support ceramic, wax and composite (Gao et al., 2015). Further, the parts built using vat photopolymerisation are inherently more prone to degradation and deformation over time, as the liquid photocurable resin generally does not have robust structural characteristics. In addition, the built parts may require special handling or tooling, which makes the process extremely expensive for some applications (AMRG, 2020).

2.2.2 Material jetting

Material jetting is an AM process in which 3D objects are built by selectively depositing the droplets of material (ISO/ASTM 52900, 2015). It was developed and patented by Objet Geometries Ltd (merged with Stratasys in 2012) in 1999 under the name of Polyjet. The development of the Polyjet process mainly derived from the standard Inkjet technology used by traditional two-dimensional (2D) printers on papers, as this process directly deposits droplets of build material onto a substrate by drop-on-demand inkjetting (Calvert, 2001; De Gans, 2004). To date, over 160 industrial AM machines using material jetting have been developed (Senvol LLC, 2020).

Material jetting is a great choice for realistic prototypes. It can provide an excellent level of details, high accuracy and smooth surface finish and supports the build of an object with multiple colours and materials. However, support material is usually required and the activated photopolymers may lose mechanical properties over time in this process (AMRG, 2020). In addition, the build material is limited to polymer and wax. Although researchers have tried direct inkjetting of nanoink suspensions of ceramic (Blazdell, 2003), metal (Ko et al., 2010) and semiconductor (Elliott et al., 2013), the suspensions have a low concentration of solid particles.

2.2.3 Binder jetting

Binder jetting is an AM process in which 3D objects are built by joining material through selectively depositing liquid binder (ISO/ASTM 52900, 2015). This process was first studied by Ely Sachs and Mike Cima at the Massachusetts Institute of Technology in 1993 and commercialised by Z Corporation (which was acquired by 3D Systems in 2012) in 1995. So far, a number of AM machine manufacturers, such as 3D Systems, ExOne, Voxeljet and EnvisionTEC, have developed over 40 industrial binder jetting AM machines (Senvol LLC, 2020).

The binder jetting process can work with a variety of materials, which include polymer, metal, ceramic, sand and composite. Using this process, 3D objects can be built with a range of different colours. However, the binder jetting process is not always suitable for functional parts because of the use of liquid binder. In addition, infiltration is usually required for the built parts to achieve sufficient strength. This can add significant time to the entire product realisation process (AMRG, 2020).

2.2.4 Powder bed fusion

Powder bed fusion is an AM process in which 3D objects are built by using laser beam, electron beam or thermal energy to selectively fuse the regions of a powder bed (ISO/ASTM 52900, 2015). The earliest powder bed fusion technology is selective laser sintering, which was developed and patented by Carl Deckard and Joe Beaman at the University of Texas at Austin in the mid-1980s. Another similar powder bed fusion technology is selective heat sintering, which uses thermal energy to replace laser beam. The idea for this technology came from Frederik Tjellesen and Anders Hartmann, the founders of Blueprinter. The technology was developed and patented by Blueprinter and was launched at Euromold in 2011. Both selective laser sintering and selective heat sintering mainly use polymer as build material. Metal is not commonly used in the two processes since there are three important and popular metal powder bed fusion techniques. They are electron beam melting, direct metal laser sintering (melting) and selective laser melting.

Electron beam melting was developed and patented by Arcam AB in 1993. Direct metal laser sintering was developed by EOS in 1994, while in 1995 the Fraunhofer Institute introduced selective laser melting. Direct metal laser sintering uses the word sintering but actually it works by melting. This is an acknowledged misnomer. A correct name of this technique is direct metal laser melting. The direct metal laser melting and selective laser melting techniques are essentially identical. Both of them can be called as LPBF to avoid confusion (King et al., 2014; Gibson et al., 2015). The electron beam melting and LPBF techniques were commercialised since 2005 by

Arcam AB and EOS GmbH, respectively. At the moment, over 390 industrial powder bed fusion AM machines have been developed.

The selective laser sintering and selective heat sintering processes benefit from requiring no additional supports, since the powder material can act as supports throughout the build process. But they need relatively long build time. Compared to these two processes, electron beam melting and LPBF are usually faster and can fabricate fully dense parts with high accuracy, strength and stiffness, which make them suitable in high-value industries such as aerospace, defence, automotive industry and medical prosthetics (Gao et al., 2015; Yap et al., 2015). However, electron beam melting and LPBF always need additional supports to compensate for high residual stress and limit the occurrence of distortion and higher power cost. They generally have relatively low efficiency of energy usage (AMRG, 2020).

2.2.5 Material extrusion

Material extrusion is an AM process in which 3D objects are built by selectively dispensing material through a nozzle or an orifice (ISO/ASTM 52900, 2015). A common material extrusion technique is fused deposition modelling, which builds layers through mechanically extruding the molten thermoplastic material onto a substrate (Gao et al., 2015). This technique was developed by Scott Crump in Stratasys Ltd in 1988 and was trademarked by this company in 1990. At present, over 240 industrial AM machines based on the material extrusion process have been manufactured (Senvol LLC, 2020).

Material extrusion is well-known as an inexpensive AM process which has been widely used in many domestic and hobby AM machines. It supports a wide variety of materials, such as polymer, composite, metal and ceramic. However, the 3D objects built by this process usually have lower dimensional accuracy and poorer surface finish compared to other AM processes (AMRG, 2020), which limits its application in the industry.

2.2.6 Directed energy deposition

Directed energy deposition is an AM process in which 3D objects are built by using focused thermal energy to melt material as it is being deposited (ISO/ASTM 52900, 2015). An important directed energy deposition technique is laser engineered net shaping, which was developed at the Sandia National Laboratories in 1995 and commercialised by Optomec in 1998. Other important techniques include laser deposition welding, 3D laser cladding, direct metal deposition, directed light fabrication and Aerosol Jet. Using these techniques, over 70 industrial AM machines have been developed and identified in the market (Senvol LLC, 2020).

The directed energy deposition process can be used with metal, ceramic and composite. It has the capability to control the grain structure to a high degree, which makes it uniquely suitable for repairing complex damaged parts (AMRG, 2020). The major drawback of the process is the poor surface finish caused by the melt pools cooling in low accuracy bands. For this reason, most directed energy deposition parts require secondary machining.

2.2.7 Sheet lamination

Sheet lamination is an AM process in which 3D objects are built by bonding sheets of material (ISO/ASTM 52900, 2015). A representative sheet lamination technique is laminated object manufacturing, which was developed by Helisys (now Cubic Technologies) in 1986 and patented by the same company in 1987. In 2000, Helisys ended operations and other companies have used proprietary version of laminated object manufacturing since then. Another important sheet lamination technique is paper 3D printing, which was developed by Conor MacCormack and Fintan MacCormack at MCor Technologies in 2003 and was commercialised by them later. So far, over 10 industrial AM machines using certain sheet lamination techniques have been manufactured by a few companies, such as EnvisionTEC, Fabrisonic, Impossible Objects and MCor Technologies (Senvol LLC, 2020).

The sheet lamination process has the advantages of ease of material handling, smooth surface finish and low material, machine and process costs (Mueller and Kochan, 1999). However, it can be used with limited types of materials. In addition, as a partially subtractive process, sheet lamination cannot obtain the same geometric complexity as other AM processes, since it may not be possible to access the internal portions of an object and remove excess material from the inside of an object (Molitch-Hou, 2018).

2.3 Laser powder bed fusion (LPBF)

LPBF, also known as direct metal laser melting or selective laser melting, is one of the powder bed fusion processes which utilises a high power-density laser beam to selectively melt and fuse metallic powders together to build near net-shape parts. The schematic of the LPBF process is illustrated in Figure 2.1 (ISO 17296-2, 2015). As can be seen from the figure, an LPBF system mainly consists of a build platform, a powder bed, a powder feeding apparatus, a recoater blade, a laser beam source and a moving mirror. The use of an LPBF system to build a 3D part mainly includes the following steps:

- A layer of powder material from the powder feeding apparatus is spread on the build platform by the recoater blade.

- A moving laser beam generated by the laser beam source and the moving mirror is applied to selectively melt and fuse the powder material to create a layer of the part.
- The build platform is moved down by one layer thickness and a new layer of powder material from the powder feeding apparatus is spread over its previous layer by the recoater blade.
- The second and third steps are repeated until the entire part is completely formed.

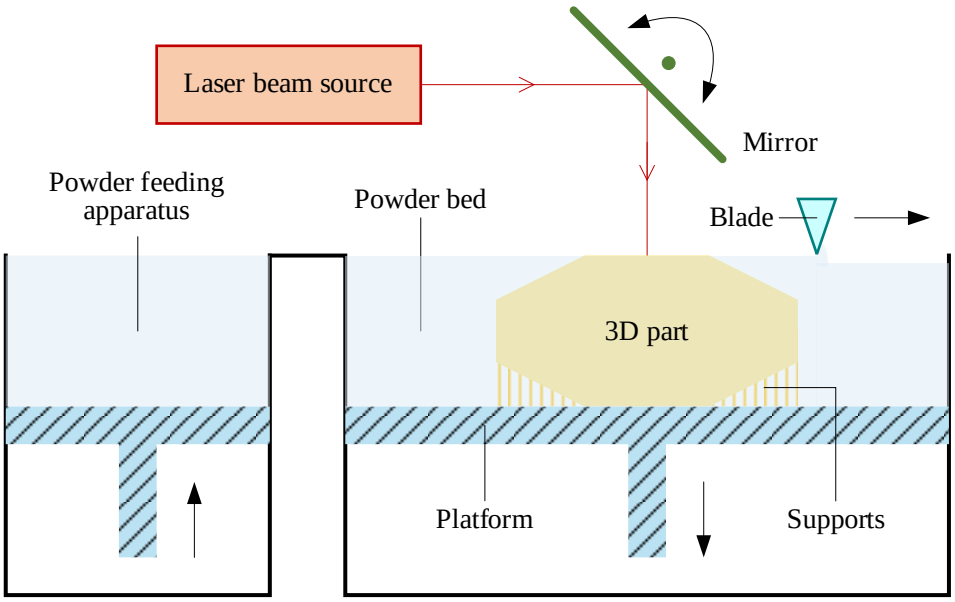


Figure 2.1 Schematic diagram of laser powder bed fusion

The LPBF process has characteristics in providing high degree of freedom for design and obtaining complex geometric shapes without additional cost, which are the common advantages of AM processes over traditional manufacturing technologies. More importantly, the LPBF process enables rapid fabrication of metal parts with near full density and high strength and stiffness directly from metallic powders without a time-consuming mould design process. This makes it an attractive technique for producing functional components in aerospace, defence, automotive and biomedical industries (Yap et al., 2015).

However, there are still many challenges for the LPBF process to be deemed mature and reliable for broader industrial applications. Representative challenges are listed as follows:

- Difficult to address the critical repeatability and reproducibility issues. At present, ensuring the repeatability of the LPBF process and the reproducibility of LPBF parts is still considered as one of the biggest challenges for facilitating a broader application of the process in the real-world industry (Dowling et al., 2020).

- Relatively limited metal materials for the process. The widely known metal materials for the LPBF process are stainless steel, tool steel, Ti6Al4V and AlSi10Mg (Yap et al., 2015). Recently, there have been developments for tungsten, zinc alloys, magnesium alloys and metal matrix composites. Even so, the metal materials that can be used in the process are still relatively limited. There are challenges faced in developing new metal materials for the process because of the difficulty to overcome the metallurgical defects like pores, lack of fusion and intermetallic formations (Sing and Yeong, 2020).
- Uncertainty and unpredictability of the process. Many applications of the LPBF process are faced with the uncertainty and unpredictability of the process due to the lack of effective process control solutions (Sing and Yeong, 2020).
- Need tedious post-processing operations. A set of tedious post-processing operations, such as heat treatment, support removal, machining, drilling, shot peening and polishing, are usually needed to be carried out on a part built by the LPBF process to achieve desired quality.
- High cost to acquire and use the process. The affordability of the LPBF process is also a critical issue for its industrial application because of the high machine, material and processing costs. For example, according to the results of a study of Baumers et al. (2013), the purchase cost of the LPBF machine EOSINT M270 is £364,406.80 and the maintenance, consumable and wire erosion costs for this machine are respectively £22,033.90 per year, £2,542.37 per year and £8,165.00 per year. The cost of using this machine and stainless steel powder (grade 17-4 PH) with a purchase cost of £78.81 per kilogram to build a part with a volume of only 96,645 mm^3 can reach £625.76.

To tackle the repeatability and reproducibility issues, new methods or techniques which can improve the design, process planning, part build and post-processing are needed to be developed (Dowling et al., 2020). An in-depth understanding of the LPBF process is essential for overcoming the metallurgical defects in the development of new metal materials. To solve the uncertainty and unpredictability issues, effective in-process and post-process monitoring systems are needed (Sing and Yeong, 2020). Optimising the design and process planning would be helpful to reduce the time and cost of processing and post-processing. The price of machines would drop drastically when they are produced in larger volumes and their related patents expire (Yap et al., 2015).

2.4 LPBF product realisation process activities

In general, the process of realising a product using the LPBF process, as shown in Figure 2.2, includes five successive activities, which are design for LPBF AM, process planning for LPBF

AM, LPBF AM part build, post-processing for LPBF AM and verification for LPBF AM (Kim et al., 2015). In this section, a brief introduction of each activity is provided.

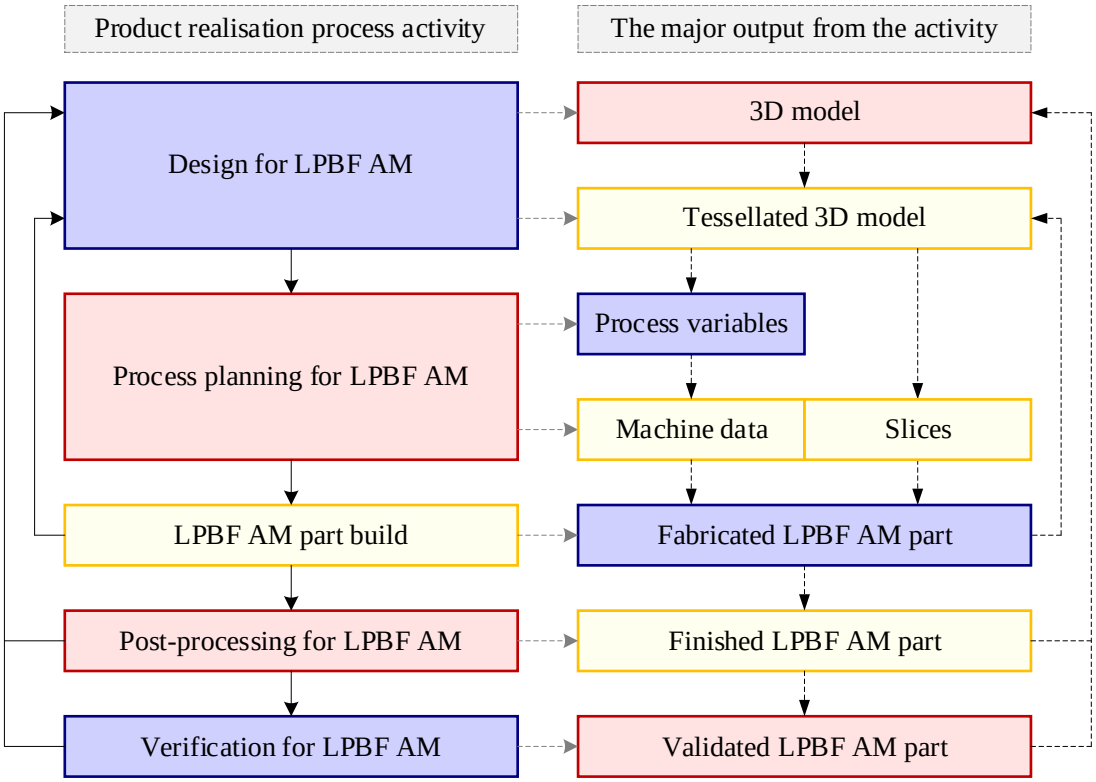


Figure 2.2 An overview of LPBF product realisation process

2.4.1 Design for LPBF AM

Design for LPBF AM refers to an activity of designing an LPBF product, in which the functional performance and other key lifecycle considerations such as manufacturability, reliability and cost of the product are optimised subjected to the capabilities of the LPBF process (Tang and Zhao, 2016; Thompson et al., 2016). The design mainly includes two tasks, which are conceptual design and detailed design. Conceptual design is the very first stage of LPBF product realisation process, in which the outline of function and the form of an LPBF product are articulated. It involves the design of interactions, experiences, processes and strategies and serves to provide a description of the proposed LPBF product in terms of concept sketches (Williams et al., 2011).

Detailed design is the stage where the design is refined and the plans, specifications and estimates are created. The most important task in this stage is geometry design (Gao et al., 2015). In general, geometry design is carried out through either CAD modelling or reverse engineering. An illustration of these two ways is shown in Figure 2.3 (ISO 17296-4, 2014). CAD modelling is

performed in a CAD system based on the results of conceptual design and outputs a tessellated 3D model of the product, which is generally encoded in the STL, OBJ (Wavefront Object), 3MF (3D Manufacturing Format) or AMF (Additive Manufacturing File) format. Reverse engineering generates a tessellated 3D model of a 3D part from its 2D image or physical model. The tessellated 3D model is also encoded in a specific format like STL, OBJ, 3MF or AMF.

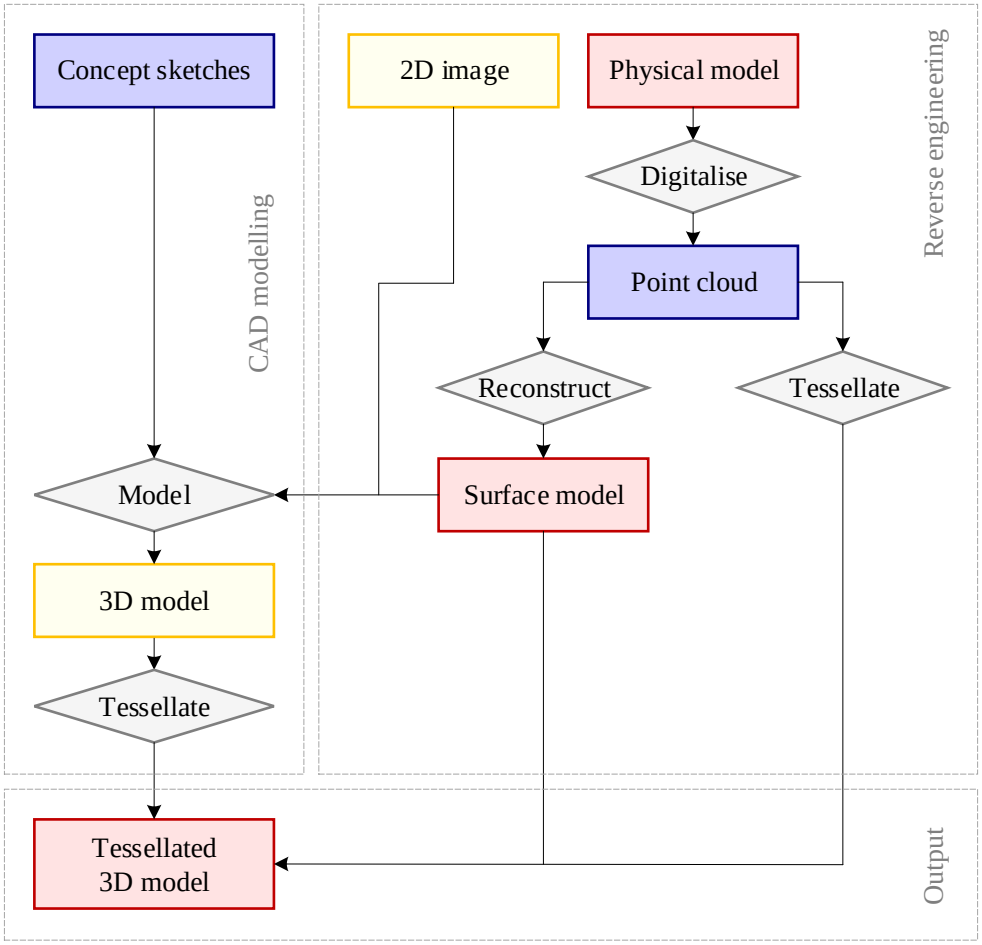


Figure 2.3 An illustration of two ways of geometry design

Apart from geometry design, detailed design also involves the design of specifications of an LPBF product and the selection of an LPBF machine to manufacture the product. Generally, the specifications to be designed can be classified from macro, micro and production levels. Macro level specifications mainly include tolerance, topology, material, colour and properties. Surface texture, material composition and porosity are micro level specifications. An important production level specification is cost model for production of the product. The details regarding the design of the specifications of an LPBF product can be found from the studies of Thompson et al. (2016) and Vaneker et al. (2020).

2.4.2 Process planning for LPBF AM

In traditional subtractive manufacturing, process planning generally involves determination of machining processes and corresponding parameters for converting a workpiece from an engineering drawing to its final form. In LPBF AM, the engineering drawing and machining processes are respectively replaced by the tessellated 3D model and LPBF processes. The aim of process planning remains the same, namely to determine specific LPBF process variables to enable efficient and accurate manufacture of a part (Kulkarni et al., 2000; Newman et al., 2015; Ahsan et al., 2015). Process planning for LPBF AM includes four successive tasks, which are build orientation determination, support generation, 3D model slicing and path planning. An illustration of these four tasks is provided in Figure 2.4. In real workshops, these tasks are generally completed in the software tools supplied by the LPBF machine manufacturers. However, the four tasks are common for all LPBF machines. Effective methods can be developed to carry out them in an integrated process planning system outside LPBF machines.

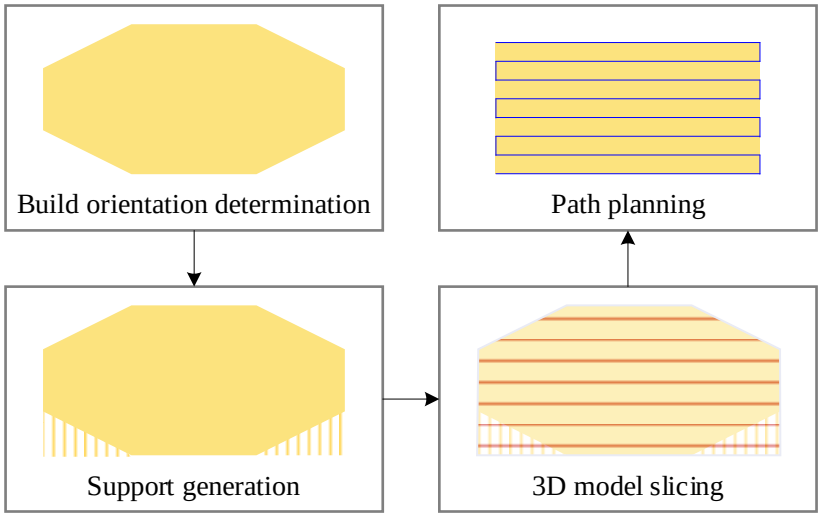


Figure 2.4 An illustration of the four tasks in process planning

Build orientation determination aims to determine a proper orientation to build an LPBF part via carrying out a comprehensive analysis of a tessellated 3D model of the part and specific production requirements and preferences on the part. Theoretically, a 3D model has infinite possible orientations in 3D space. One of the feasible approaches for build orientation determination is to generate a certain number of alternatives from the infinite possible orientations first and then select an optimal alternative from the generated alternatives (Kulkarni et al., 2000). Based on this, build orientation determination can be divided into ABO generation and OBO selection.

In build orientation determination, the tessellated 3D model is the output from the design activity. The production requirements and preferences for ABO generation and OBO selection are specified by designers or process planners on the basis of certain build orientation attributes (i.e. the factors influenced by build orientation). According to the international standard ISO 17296-3 (2014) and the survey of Di Angelo et al. (2020), the main build orientation attributes include:

- Support structure. In the LPBF process, supports are needed to sustain the overhanging areas to resist deformation or collapse or to reduce part distortion caused by thermal gradients. It is intuitive that different build orientations require different amount of supports. An ideal build orientation should need as few supports as possible under the premise of ensuring successful build, as the use of supports leads to an increase of the build time, build cost, post-processing time and post-processing cost and the removal of supports could be detrimental to the surface quality of the support contact areas.
- Part property. The mechanical properties of an LPBF part, such as strength, elongation, hardness, and the physical properties of an LPBF part, such as residual stress, flexural modulus and fatigue performance, are generally influenced by the build orientation of the part. For example, the strength of a part built by the LPBF process is always anisotropic. In general, the tensile strength and yield strength in horizontal orientation are respectively greater than the tensile strength and yield strength in vertical orientation.
- Part accuracy. Part accuracy generally includes dimensional error, geometric error and volumetric error. The build orientation of an LPBF part directly affects the shrinkage, curling and distortion of the part, which are the major factors influencing part accuracy.
- Surface quality. Build orientation has an important influence on surface quality for the LPBF process due to its layer upon layer building manner and the use of supports. As an example, planes or surfaces that are parallel or perpendicular to build orientation would have smaller surface roughness than those that have an angle with build orientation. Declining faces would be more seriously influenced by the staircase effect.
- Build time. Build time mainly consists of layer scanning time, layer preparation time and recoating time. For the LPBF process, build orientation directly influences build time because of the layer by layer nature of the process and the use of supports.
- Build cost. Build cost calculates all resources required in building an LPBF part, such as machine, material, energy and labour. It is also affected by the build orientation of the part, because different build orientations may cause different support volume, material consumption, energy consumption and build time.

- Post-processing time: Post-processing time mainly contains property enhancement time, support removal time, accuracy improvement time and surface quality improvement time. Build orientation also affects the post-processing time, since it influences the support structure, part property, part accuracy and surface quality.
- Post-processing cost: Post-processing cost calculates all resources needed in post-processing operations on a part built by the LPBF process. It is also affected by build orientation, as different build orientations may result in different support structure, part accuracy, surface quality, part property and post-processing time.

From the overall value chain, part property, part accuracy and surface quality should be considered emphatically in build orientation determination, since they directly reflect the quality of a built part and satisfying the quality requirements is the most basic condition for practical application of the LPBF process. It is also of necessity to consider support structure as this attribute has an influence on the processing time and material and processing costs. The main purpose of taking into account the remaining attributes is to improve production efficiency and reduce production cost under the premise of ensuring product quality. In the existing build orientation determination methods, all of the eight attributes above have been covered. This will be described in detail in Section 2.5.

The second task in process planning for LPBF AM is support generation, which aims to determine the minimum supports needed to successfully build an LPBF part via performing a geometric analysis of a tessellated 3D model of the part in a specified orientation. In support generation, the tessellated 3D model in a specified orientation is the output of the build orientation determination task.

Supports can be classified into external supports and internal supports according to the areas to be supported. As depicted in Figure 2.5, external supports are used to sustain overhangs and internal supports are used to sustain the top areas of a hollow 3D model. Theoretically, areas of the tessellated 3D model requiring external supports are the areas whose surface normal is pointing downwards with respect to the build platform. However, as illustrated in Figure 2.6, some overhangs can be built without supports in the LPBF processes since the width of the accumulated material can withstand them (Kulkarni et al., 2000). According to the geometric shape, supports for the LPBF process can be classified into lattice supports, cellular supports, unit cell supports, pin supports and tree supports (Jiang et al., 2018; Zhu et al., 2020). Each shape of supports has its characteristics and best conditions of use. In practice, the shape of supports can be selected according to the geometry of the 3D model and the used LPBF machine.

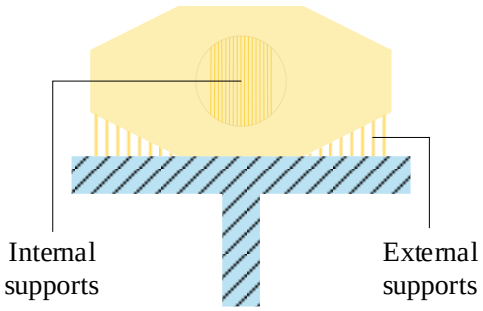


Figure 2.5 An illustration of two types of supports

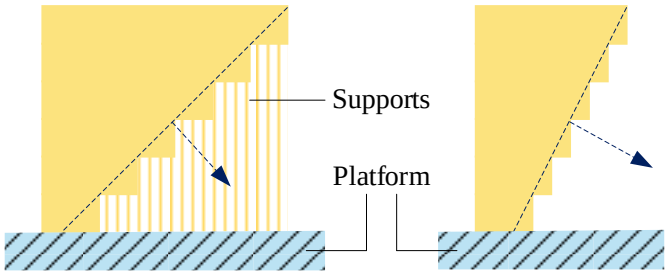


Figure 2.6 An illustration of two types of overhangs

Supports can be generated either from a tessellated 3D model or an original 3D model (Jiang et al., 2018). Using a tessellated 3D model, it is easy to determine facets that need supports. For example, facets whose normal vectors are pointing downwards with respect to the build platform may require external supports. For the generation of supports based on the original 3D model, a Gaussian map of surfaces of the model can indicate areas requiring supports, but computation of the Gaussian map of a freeform surface model is extremely complicated (Kulkarni et al., 2000).

The third task in process planning for LPBF AM is 3D model slicing, which transforms process planning from the 3D model domain to a 2.5D (two-point-five-dimensional) layer domain. The 3D model slicing task involves intersecting a 3D model with a horizontal plane. Its input is a tessellated 3D model of a part with supports in a specified orientation, and its output, as depicted in Figure 2.7, includes the thickness of individual layers and the geometry of the contour to be accumulated. Each individual layer is called a slice. The output slice information is usually encoded in a proprietary file format, such as the SLC (Stereo Lithography Contour), CLI (Common Layer Interface) or HPGL (Hewlett-Packard Graphics Language) format. Apart from tessellated 3D model, 3D model slicing can also be carried out on an original CAD model or a reverse engineering model (Zhao and Guo, 2020).

A trivial strategy for 3D model slicing is uniform planar slicing in single direction (see Figure 2.8(a)), in which the thickness is identical for all layers and the 3D model is truncated in a

single direction. Historically, this is the first slicing strategy adopted in AM and continues to be the norm in the industry. Even though the strategy is simple, efficient and robust, it will result in a loss of control over the accuracy of the built part, because it neglects the actual geometry of the 3D model (Kulkarni et al., 2000). A modified strategy is to truncate the 3D model in a single direction with variable layer thickness (see Figure 2.8(b)). This strategy is helpful to improve the accuracy of the built part, but it generally requires some complex and inefficient algorithms that are sometimes not robust.

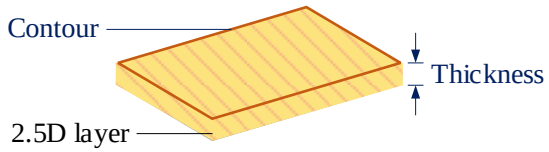


Figure 2.7 An illustration of two elements of 3D model slicing

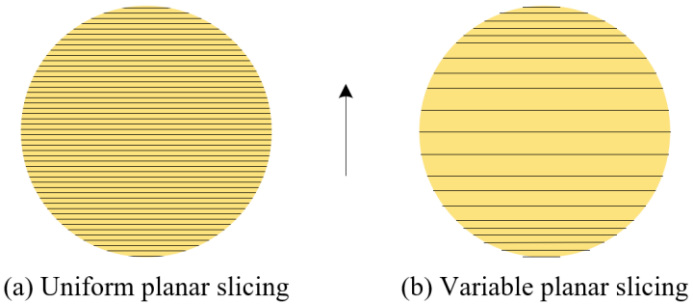


Figure 2.8 An illustration of two common slicing strategies

In the literature, many more complicated slicing strategies which consider the actual geometry of the 3D model are available. Representative ones are uniform planar slicing in multiple directions, variable planar slicing in multiple directions, uniform curved slicing in single direction, variable curved slicing in single direction, uniform curved slicing in multiple directions and variable curved slicing in multiple directions. Compared with the two strategies in Figure 2.8, these strategies are generally not very efficient. A detailed introduction and analysis of these strategies can be found from the survey of Zhao and Guo (2020).

The last task in process planning for LPBF AM is path planning. It belongs to a pure layer domain task. Path planning aims to determine the laser scanning path and process parameters for building each layer from slices of a tessellated 3D model of a part with supports.

A very important step in path planning is to determine a proper 2D path pattern. The determination mainly depends on the used slicing strategy. If a planar slicing strategy is used, alterna-

tive path patterns mainly include raster path, zigzag path, grid path, spiral path, contour path and hybrid path, as illustrated in Figure 2.9 (Zhao and Guo, 2020). The raster path is simple, efficient and robust for any boundaries, but the build accuracy of each layer is poor. The zigzag path can link separate parallel lines into a single continuous pass and therefore reduces the build time, but it will result in poor surface quality of the built part. The grid path is capable of improving the bonding strength between adjacent layers and thus improve the mechanical properties of the built part. However, there are lots of path passes. The spiral path is generally used in numerical control machining. It is only suitable for certain slices. The contour path is used via offsetting the contours of a slice. The hybrid path combines two or more single paths to achieve multiple advantages concurrently. Among the six path patterns, the most used ones are contour path and raster path due to their simplicity and robustness. If a curved slicing strategy is adopted, path planning will become more complicated. The available path patterns include directional parallel path, contour parallel path and space-filling curve path. A detailed analysis of these path patterns can be found from the survey of Zhao and Guo (2020).

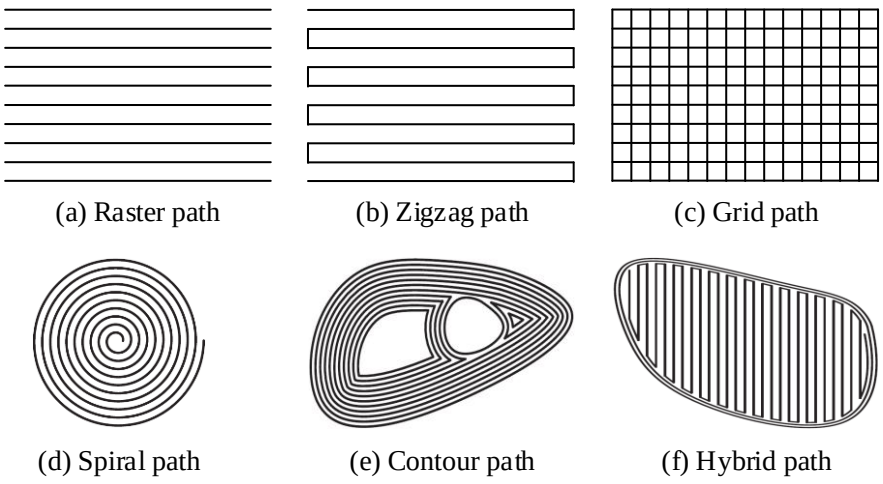


Figure 2.9 An illustration of six common path patterns

Another very important step in path planning is to determine a group of optimal process parameters. Among all process parameters that need to be determined, layer thickness, hatch spacing, laser power and laser scanning velocity are four critical process parameters. The volumetric energy density calculated by them is an important indicator to measure the influence of process parameters on part quality. For optimisation design of process parameters, many existing methods first achieve different combinations of process parameters according to the technical data of the used LPBF machine, then carry out certain experiments to study the effect of each combina-

tion on part quality, and finally determine the optimal process parameters according to the results of the experiments (Shipley et al., 2018).

2.4.3 LPBF AM part build

LPBF AM part build is an activity of using a specific LPBF machine to build an LPBF part. This activity consists of three stages, which are preparation, build and teardown. In the preparation stage, the designed geometry and specifications and determined process variables are imported into the used LPBF machine and converted into machine-interpretable information. Then the material is prepared, and the machine is set up. When all the preparations are completed, the machine starts to build the part. The build stage is generally associated with monitoring and controlling the build environment. In the teardown stage, the built part is removed from the machine and cleanup is carried out.

2.4.4 Post-processing for LPBF AM

A part built by the LPBF process usually needs post-processing to satisfy the designed specifications. For an LPBF part, post-processes may include property enhancement, support removal, accuracy improvement, surface finish improvement and esthetic improvement (Kim et al., 2015). Property enhancement using heat treatment techniques (e.g. hot isostatic pressing, annealing) is helpful to improve the mechanical and physical properties of the parts (Wauthle et al., 2015). In general, property enhancement is carried out before support removal to relieve residual stresses. Support removal can be carried out using certain instruments or tools, such as wire-electrical discharge machining instrument and bandsaw. The geometric accuracy of a part built by the LPBF process is generally poorer than that manufactured by conventional machining processes and the dimensional variability is also common. To reduce such variability, accuracy improvement using adaptive raster milling, sharp edge contour machining and drilling is usually required. In post-processing for LPBF AM, surface finish improvement is also often needed, because poor surface finish may be induced from the staircase effect of the process and the use of supports. Methods for this improvement generally include shot peening, painting and hardening. Esthetic improvement is sometimes carried out to improve the appearance of the built part. Common esthetic improvement techniques are painting, priming and polishing.

2.4.5 Verification for LPBF AM

Verification for LPBF AM is the last activity in LPBF product realisation process. This activity aims to qualify the finished LPBF part and determine whether the design requirements have

been satisfied via measurement of errors, surface texture, defects and part properties. Common measuring and inspection instruments for errors are industrial computed tomography scanner, 3D optical scanner and coordinate measuring machine. Surface texture inspection mainly includes the measurement of profile topography and areal topography. Profile topography inspection is commonly carried out using contact stylus. Technologies or tools for areal topography measurement are more diverse, which include confocal microscopy, atomic force microscopy, focus variation microscopy, coherence scanning interferometry, conoscopic holography and elastomeric sensor (Townsend et al., 2016). Inspection of defects is usually conducted using non-destructive evaluation techniques, such as remote visual inspection, thermographic inspection, stereomicroscope inspection, ultrasonic inspection and acoustic emission inspection. Part properties can be categorised into mechanical properties (e.g. strength, elongation, hardness), physical properties (e.g. residual stress, flexural modulus, fatigue performance), chemical properties (e.g. flammability, toxicity, corrosion) and thermal properties (e.g. softening temperature, melting point, thermal conductivity). The methods for testing part properties are documented in the international standards ISO 17296-3 (2014) and ISO/ASTM 52904 (2019).

2.5 Build orientation determination methods

An ideal method for build orientation determination is preferably a standardised method for practical applications. The latest international standards related to AM design and process planning ISO/ASTM 52911-1 (2019) and ISO/ASTM 52911-2 (2019) respectively provides design guidelines for LPBF of metals and LPBF of polymers, and do not yet contain a practical method for determination of build orientation. To provide an effective tool for build orientation determination, many methods have been presented during the past three decades. As depicted in Figure 2.10, the methods can be categorised into ABO generation methods and OBO selection methods based on their focuses. In this section, a review of each category of methods is provided.

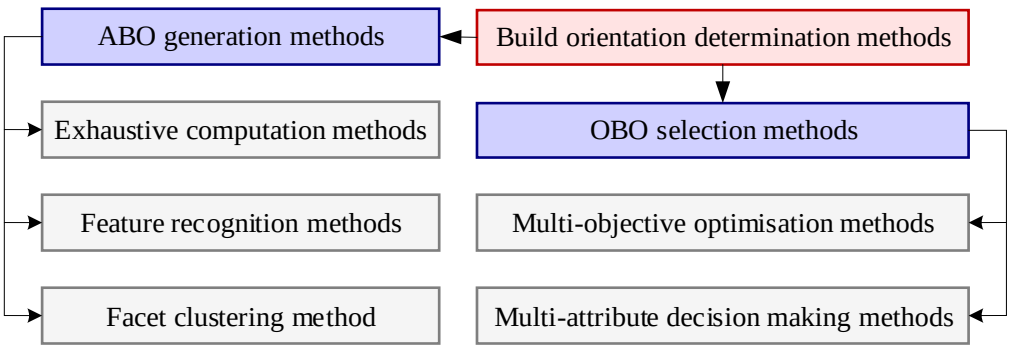


Figure 2.10 Two categories of build orientation determination methods

2.5.1 Alternative orientation generation methods

To achieve generation of the ABOs for an AM part, there are a number of methods available. These methods can be classified into exhaustive computation methods, feature recognition methods and facet clustering method on the basis of their fundamental principles.

The exhaustive computation methods firstly rotate the 3D model in 3D space with a specific step size. Specific algorithms are then leveraged to compute the quality degree of the orientation corresponding to each step, and a certain number of ABOs are obtained according to the results of computation. Representative exhaustive computation methods are the methods presented by Xu et al. (1997), Hur and Lee (1998), McClurkin and Rosen (1998), Masood et al. (2000; 2003), Rattanawong et al. (2001), Paul and Anand (2011; 2015), Ezair et al. (2015), Delfs et al. (2016), Chowdhury et al. (2018), Galicia and Benes (2018), Griffiths et al. (2019), Jiang et al. (2019), Fritz and Kim (2020), Nguyen and Choi (2020), Ulu et al. (2020) and Wang and Qian (2020).

The exhaustive computation methods generally need to consume a lot of time in the calculation of meaningless orientations, since there is a contradictory issue in them: How to set a suitable rotation step size? If the rotation step size is set too large, the number of computations for the entire generation process can be reduced, but it is very likely that the true OBO will be missed; if it is set too small, the chance of missing the true OBO is greatly reduced, but it will greatly increase the number of computations in the entire generation process; If it is set randomly, an OBO could be generated in a short time. But it is more likely to take a very long time. For this reason, most of the exhaustive computation methods are difficult to apply in practical build orientation determination due to their high computational cost.

The feature recognition methods firstly decompose the STL model into a certain number of surface fragments which consist of adjoining planar triangular facets. They then recognise which predefined shape feature each surface fragment belongs to. According to the feature type of each surface fragment, the ABOs of the surface fragment are generated. The ABOs of the part are obtained through combining the ABOs of all surface fragments of its STL model. From this principle, it is not difficult to observe that the feature recognition methods focus on a finite set of orientations. They will not spend time on computation of meaningless orientations like the exhaustive computation methods. However, how to decompose an STL model to form predefined shape features is difficult due to the lack of topological information in the model. In addition, automatic recognition of shape features from a complex 3D model with multiple overlapping shape features is a challenging task. Last but not the least, the methods are not applicable for freeform surface models, as it is difficult to define the shape features for such models.

Representative feature recognition methods include the methods presented by Cheng et al. (1995), Frank and Fadel (1995), Lan et al. (1997), Alexander et al. (1998), Pham et al. (1999), Xu et al. (1999), West et al. (2001), Byun and Lee (2006), Zhang et al. (2016) and Al-Ahmari et al. (2018). The methods of Alexander et al. (1998), Xu et al. (1999) and Byun and Lee (2006) generate the ABOs of an AM part based on the convex hull of the STL model. They are simple, intuitive and easy to implement. However, all of these methods have an accuracy issue since the convex hull is not the accurate STL model. In the method of Lan et al. (1997), ABOs are generated as the orientations of the base planes of the STL model, which is simpler and more intuitive. However, this method is difficult to achieve desired results for the STL models with curved surface features, as the base planes of such STL models are generally difficult to find. To solve this issue, the methods of Cheng et al. (1995), Frank and Fadel (1995), Pham et al. (1999) and West et al. (2001) introduce the concept of AM feature. They firstly determine the correspondence between each type of AM features and build orientations based on production experience. According to the determined correspondence, the rules for generation of ABOs are then designed. Finally, the AM features in the STL model are recognised manually and the ABOs of the part are obtained via rule-based reasoning. These methods provide a viable idea for computer-aided build orientation determination, but they do not provide a clear definition and specific classification of AM features. Aiming at this issue, the method of Zhang et al. (2016) presents a definition of AM features and classifies AM features into cylinders, planes, cones and structural units. The generation methods based on AM features then become more reliable. However, this classification is rather simple. Many STL models cannot be represented using the classified AM features. Based on the method of Zhang et al. (2016), the method of Al-Ahmari et al. (2018) implements automatic generation of ABOs for AM parts, but this method has not yet addressed automatic recognition of AM features from an STL model with multiple overlapping AM features.

The facet clustering method presented by Zhang et al. (2019) applies the *k*-means clustering algorithm (Hartigan and Wong, 1979) and the Davies-Bouldin index (Davies and Bouldin, 1979) to automatically divide the STL model of an AM part into a certain finite number of clusters of meaningful discrete planar triangular facets and directly generates the ABOs of each cluster. The ABOs of the part are achieved via combining and refining the ABOs of all clusters of its STL model. Compared to the feature recognition methods, this method does not need topological information and avoids shape feature recognition. Most importantly, it is applicable for both regular and freeform surface models. However, it is found from practical applications that the method suffers from the following issues:

- The ABO generation results of the method are unstable, because the initial *k* planar triangular facets in it need to be randomly selected and different *k* initial facets would obtain different clustering results.
- The efficiency of the method is still an issue for an STL model with a large number of facets, since it will take a lot of time to determine the value of *k* and to calculate the angles between each normal vector cluster and the remaining normal vectors.
- The method could generate unreasonable facet clustering results with facet clusters of varying probability density, because the probability density function in it is assumed to obey the *k* Gaussian distributions.

2.5.2 Optimal orientation selection methods

According to the techniques applied in OBO determination, existing methods for selection of the OBO for an AM part can be grouped into multi-objective optimisation (MOO) methods and MADM methods.

An MOO method generally uses certain optimisation techniques to search a build orientation enabling one or more specific attributes to be optimal from an infinite number of possible orientations or a certain number of ABOs. The research of MOO methods for build orientation determination has gained importance and popularity within the academia during the past few decades. A large number of MOO methods have been presented in this period. A brief summarisation of a set of representative methods is provided in Table 2.1.

As can be seen from Table 2.1, the main difference of MOO methods lies in the targeted AM processes, the optimisation techniques and the optimised attributes:

- Targeted processes. Some MOO methods are theoretically applicable for all processes, while each of the remaining methods is proposed for one or more specific processes, including stereolithography, selective laser sintering, fused deposition modelling and LPBF.
- Optimisation techniques. Most MOO methods are based on exhaustive search or applied the genetic algorithm. Other optimisation techniques include the non-dominated sorting genetic algorithm II, trust region method, particle swarm algorithm, extreme learning machine, principal component analysis-based algorithm, surrogate model, Taguchi method, bacterial foraging algorithm and electromagnetism-like mechanism algorithm.
- Optimised attributes. Most MOO methods aim to optimise one or two attributes. Some methods can optimise three or four attributes simultaneously. Only a few methods considered the optimisation of more than four attributes.

Table 2.1 A brief summarisation of representative MOO methods

MOO method	Targeted processes	Optimisation techniques	Optimised build orientation attributes							
			SS	PP	PA	SQ	BT	BC	PT	PC
Xu et al. (1997)	SLA	Exhaustive search	●		●		●			
Hur and Lee (1998)	SLA	Exhaustive search	●		●		●			
McClurkin (1998)	SLA	Exhaustive search			●	●	●			
Masood et al. (2000)	FDM	Exhaustive search			●					
Hur et al. (2001)	SLS	Genetic algorithm					●			
Rattanawong et al. (2001)	FDM	Exhaustive search			●					
Masood et al. (2003)	FDM	Exhaustive search			●					
Pandey et al. (2004)	FDM	NSGA-II				●	●			
Thrimurthulu et al. (2004)	FDM	Genetic algorithm				●	●			
Byun and Lee (2005)	3 processes	Genetic algorithm				●	●			
Kim and Lee (2005)	SLA	Genetic algorithm					●		●	●
Singhal et al. (2005)	SLA	Trust region method				●				
Ahn et al. (2007)	3 processes	Genetic algorithm				●				
Canellidis et al. (2009)	SLA	Genetic algorithm				●	●			
Singhal et al. (2009)	SLA, SLS	Trust region method	●			●	●			
Nezhad et al. (2010)	SLA	Genetic algorithm	●			●	●			
Padhye and Deb (2011)	SLS	NSGA-II, PSA				●	●			
Paul and Anand (2011)	SLA, SLS	Exhaustive search			●					
Strano et al. (2011)	SLS	Genetic algorithm				●		●		
Phatak and Pande (2012)	3 processes	Genetic algorithm				●	●	●		
Li and Zhang (2013)	Not specified	Genetic algorithm			●		●			
Zhang and Li (2013)	Not specified	Genetic algorithm			●					
Ezair et al. (2015)	Not specified	Exhaustive search	●							
Paul and Anand (2015)	Not specified	Exhaustive search	●		●					
Zhang et al. (2015)	SLA, FDM	ELM	●							
Ahsan and Khoda (2016)	Not specified	Genetic algorithm					●			
Delfs et al. (2016)	SLS	Exhaustive search				●				
Luo and Wang (2016)	Not specified	PCABA			●					
Brika et al. (2017)	LPBF	Genetic algorithm	●	●		●	●	●		
Zhang et al. (2017)	Not specified	Genetic algorithm					●	●		
Chowdhury et al. (2018)	Not specified	Exhaustive search	●		●		●			
Galicia and Benes (2018)	FDM	Exhaustive search					●			
Huang et al. (2018)	FDM	NSGA-II				●	●			
Jaiswal et al. (2018)	FDM	Surrogate model			●					
Pereira et al. (2018)	Not specified	PSA	●		●					
Cheng and To (2019)	LPBF	PSA	●	●						
Golmohammadi (2019)	FDM	Taguchi method	●			●				
Griffiths et al. (2019)	LPBF	Exhaustive search						●		
Jiang et al. (2019)	FDM	Exhaustive search	●							
Raju et al. (2019)	FDM	PSA, BFA		●		●				
Fritz and Kim (2020)	LPBF	Exhaustive search					●	●		
Matos et al. (2020)	FDM	ELMA	●		●	●	●			
Nguyen and Choi (2020)	FDM	Exhaustive search		●						
Shen et al. (2020)	FDM	PSA	●							
Ulu et al. (2020)	Not specified	Exhaustive search			●					
Wang et al. (2020)	FDM	Genetic algorithm	●				●			
Wang and Qian (2020)	Not specified	Exhaustive search	●							

Notes: SS: Support structure; PP: Part property; PA: Part accuracy; SQ: Surface quality; BT: Build time; BC: Build cost; PT: Post-processing time; PC: Post-processing cost; SLA: Stereolithography; FDM: Fused deposition modelling; SLS: Selective laser sintering; NSGA-II: Non-dominated sorting genetic algorithm II; 3 processes: Stereolithography, selective laser sintering, fused deposition modelling; PSA: Particle swarm algorithm; ELM: Extreme learning machine; PCABA: Principal component analysis-based algorithm; BFA: Bacterial foraging algorithm; ELMA: Electromagnetism-like mechanism algorithm

An ideal MOO method for OBO selection should be an effective, stable and efficient method that is applicable for as many different AM processes as possible and can simultaneously optimise as many different attributes as possible. However, optimising multiple attributes at the same time may need a large number of iterations, which is likely to produce a large number of Pareto-optimal build orientations and would greatly extend the convergence time of the used optimisation algorithm (Ancau and Caizar, 2010). These are unbearable for build orientation determination in engineering practice (Zhang et al., 2019). For this reason, most of the software tools supplied by AM machine manufacturers recommend or only support the optimisation of one or two attributes in determination of a desirable build orientation.

In an MADM method, certain decision making techniques are used to estimate the overall score of each ABO in the case of concurrently considering its multiple attributes. All ABOs are ranked according to the overall scores and the OBO is selected based on the ranking. Representative MADM methods are the methods proposed by Pham et al. (1999), West et al. (2001), Byun and Lee (2006), Chen et al. (2008), Zhang et al. (2016), Ransikarbum and Kim (2017), Qie et al. (2018) and Yu et al. (2019).

The method of Pham et al. (1999) offers a decision support tool based on the weighted arithmetic average operator to help stereolithography users in part orientation. In this tool, orientated features, problematic features, overhanging area, support volume, build time and build cost are considered concurrently to recommend a desirable orientation. The method of West et al. (2001) was presented to aid stereolithography users in selecting proper values of part orientation, layer thicknesses and recoating variables. In this method, a deviation function is used to obtain a balance of objectives specified by geometric error, surface roughness and build time. In the method of Byun and Lee (2006), the best orientation for the stereolithography, selective laser sintering, fused deposition modelling and laminated object manufacturing processes is selected via MADM based on the weighted arithmetic average operator. The attributes considered in the selection are surface roughness, build time and part cost. In the method of Chen et al. (2008), seven attributes affected by build orientation in automated layer-based machining, i.e. inaccessible volume, base plane size, skewness of centre of gravity, height of centre of gravity, support contact area, number of layers and material waste, are formulated on the basis of the STL model of a part and represented as fuzzy variables. A fuzzy MADM method is applied to rank alternative orientations and the optimal orientation is determined according to the ranking. The method of Zhang et al. (2016) adopts an integrated MADM model (Zhang and Bernard, 2014) to select the best orientation from ABOs for AM. In this method, favourableness of features, support volume, mechanical

properties, part accuracy, surface quality, build time, build cost and post-processing are considered at the same time. The method of Ransikarbum and Kim (2017) provides an analytic hierarchy process framework for selection of the part orientation in fused deposition modelling. In this framework, a trade-off among the support volume, mechanical properties, part accuracy, surface quality, build time and build cost is analysed and compared to achieve the OBO. In the method of Qie et al. (2018), a feedback MADM model is established to assist build orientation determination in the stereolithography process. This model considers support volume, surface roughness and build time and uses the ordered weighted average operator for aggregation. In the method of Yu et al. (2019), a feedback decision-making model is constructed to realise personalised design of build orientation for stereolithography and fused deposition modelling. Support volume, geometric error, surface roughness, build time and part cost are considered in the model.

Based on the description above, a brief summarisation of representative MADM methods is provided in Table 2.2.

Table 2.2 A brief summarisation of representative MADM methods

MADM method	Targeted processes	Decision making technique	Considered build orientation attributes							
			SS	PP	PA	SQ	BT	BC	PT	PC
Pham et al. (1999)	SLA	WAA operator	●				●	●		
West et al. (2001)	SLA	Deviation function			●	●	●			
Byun and Lee (2006)	4 processes	WAA operator				●	●	●		●
Chen et al. (2008)	Not specified	Fuzzy method	●				●	●		
Zhang et al. (2016)	Not specified	Integrated model	●	●	●	●	●	●	●	●
Ransikarbum (2017)	FDM	Hierarchical analysis	●	●	●	●	●	●		
Qie et al. (2018)	SLA	Feedback model	●			●	●			
Yu et al. (2019)	SLA, FDM	Feedback model	●		●	●	●	●		●

Notes: SS: Support structure; PP: Part property; PA: Part accuracy; SQ: Surface quality; BT: Build time; BC: Build cost; PT: Post-processing time; PC: Post-processing cost; SLA: Stereolithography; WAA: Weighted arithmetic average; 4 processes: Stereolithography, selective laser sintering, fused deposition modelling, laminated object manufacturing; FDM: Fused deposition modelling; OWA: Ordered weighted average

It can be seen from the table that different decision making techniques are used and various sets of attributes are considered in these methods. Like the MOO methods, the MADM methods also determine the OBO by considering certain build orientation attributes. The two categories of methods involve roughly the same types of attributes, but they are essentially different. The core of the MOO methods is optimisation. That is, the OBO is obtained through optimising the values of attributes. The core of the MADM methods is aggregation. That is, the OBO is determined via aggregating the values of attributes. Aggregation has the following two advantages over optimisation (Zhang et al., 2016):

- The number of the optimal solutions obtained through aggregation is generally one or many when the number of attributes increases. If several solutions have the same highest score, any one of them can be selected as the optimal solution.
- The time required for aggregation generally increases in a linear relationship with the number of attributes and ABOs.

From the advantages above, it is not difficult to conclude that an MADM method is generally better suited than an MOO method in practical application. However, the existing MADM methods still have two main issues:

- The complex relationships among the attributes of ABOs have not been captured. The different attributes used for OBO selection are usually not independent of each other, but are interrelated. But all of the methods in Table 2.2 assume that the attributes are independent of each other when aggregating their values (Zhang et al., 2019).
- Negative influence from the deviation of values of attributes has not been reduced. The values of attributes are generally obtained through theoretical calculation, simulation estimation or expert evaluation. It is usually impossible to ensure the absolute objectivity of these ways, which means that there would be deviation in one or several values of attributes. To determine reasonable OBO under such circumstance, it is of necessity to reduce the effect of deviation on the aggregation result. But none of the methods in Table 2.2 can achieve this.

2.6 Research gaps

As can be concluded from the review and analysis of the existing ABO generation methods in Section 2.5.1, the existing exhaustive computation methods are difficult to apply in engineering practice because of their high computational cost. The existing feature recognition methods can address this limitation, but how to decompose an STL model to construct predefined shape features is difficult for these methods due to the lack of topological information in the model. In addition, the methods only work for regular surface models and are not applicable for freeform surface models. The existing facet clustering method does not need topological information and can be applied to both regular and freeform surface models. From this point of view, a facet clustering method could be more advantageous for the generation of ABOs than an exhaustive computation method and a feature recognition method.

From the review and analysis of the existing OBO selections methods in Section 2.5.2, it can be concluded that using an MOO method to optimise multiple build orientation attributes at the same time is impractical for engineering practice, as it is likely to generate a large number of Pa-

reto-optimal build orientations and will greatly extend the convergence time of the used optimisation algorithm. Compared to an MOO method, an MADM method always outputs one or many OBOs when the number of attributes increases. Further, the time required for an MADM method generally increases in a linear relation with the number of attributes and ABOs. Thus an MADM method could be better suited than an MOO method in engineering practice.

Based on the conclusions above, a combination of a facet clustering method and an MADM method could be a relatively ideal approach for build orientation determination for AM. However, the existing facet clustering method and MADM methods still have the following limitations:

- The existing facet clustering method may produce unstable results, would not be efficient for an STL model with a large number of facets and could generate unreasonable results for clusters of varying probability density.
- The existing MADM methods have not captured the complex relationships among the attributes of build orientation and have not reduced the negative effect of the deviation of values of attributes on the aggregation result.

In this thesis, LPBF is chosen as the focused AM process and the automatic determination of part build orientation for LPBF AM is studied based on the combination above. The reason for choosing the LPBF process is that LPBF is one of the most suitable AM processes for producing mechanical components. Mechanical products are usually made of metal materials. Users generally have high quality and functional requirements for them. Although there are a number of AM processes supporting metal materials, the components built by most of these processes are difficult to satisfy the high quality and functional requirements. Conversely, the LPBF process allows rapid fabrication of fully dense metal components with high accuracy, strength and stiffness directly from metallic powders without the time-consuming mould design process. It is an attractive technology for producing high quality functional components in industrial applications (Sing and Yeong, 2020). The motivations of the study of automatic determination of part build orientation for LPBF AM are described as follows:

- In response to the limitations of the existing facet clustering method, the accelerated HDBSCAN* algorithm (McInnes and Healy, 2017) is introduced to develop a novel facet clustering method for automatic generation of the ABOs of an LPBF part. This algorithm is an unsupervised learning algorithm to produce clusters of a dataset and is an accelerated version of the original HDBSCAN* algorithm (Campello et al., 2015). It is capable of improving the classic *k*-means clustering algorithm (Hartigan and Wong, 1979) at the aspects of requiring assignment of the number of clusters, producing unstable locally optimal clustering results,

sensitive to noise and implicit assumption that clusters have Gaussian distributions. The algorithm can also make the computational scalability of the original algorithm comparable in efficiency to mainstream clustering algorithms. Because of such characteristics, the developed novel facet clustering method can always produce stable results, provide satisfying efficiency for an STL model with a large number of facets and work well with facet clusters of varying probability density. The details of this method will be explained in Chapter 3. Illustration and evaluation of the method will be reported in Chapter 6.

- Aiming at the limitations of the existing MADM methods, the fuzzy number (Zadeh, 1965), Muirhead mean (MM) and geometric Muirhead mean (GMM) operators (Muirhead, 1902), power average (PA) operator (Yager, 2001) and Archimedean t-norm and t-conorm (Klement et al., 2000) are introduced to construct a set of fuzzy AOs for aggregating the values of attributes of ABOs. Based on the constructed fuzzy AOs, a novel MADM method for automatic selection of the OBO of an LPBF part is proposed. The fuzzy number is a mathematical tool that can normalise the values of attributes into the numbers in [0, 1] to make them easy to process. The MM and GMM operators are two all-in one AOs for capturing the interrelationships of aggregated arguments, since they are applicable whenever all aggregated arguments are independent of each other, or there are interrelationships between two aggregated arguments, or three or more aggregated arguments. The PA operator is an AO that can reduce the negative influence of unreasonable argument values on the aggregation result. The Archimedean t-norm and t-conorm are a set of mathematical operations that can be used to develop general and flexible operational laws for fuzzy numbers. Benefiting from a combination of the fuzzy number, MM and GMM operators, PA operator and operational laws based on Archimedean t-norm and t-conorm, the proposed novel MADM method can capture the interrelationships among the attributes of ABOs in the aggregation of the values of attributes, while it can also reduce the negative influence of the deviation of attribute values on the aggregation result. The details of the constructed fuzzy AOs and the proposed method will be respectively explained in Chapter 4 and Chapter 5. Illustration and evaluation of the method will be reported in Chapter 6.

2.7 Summary

This chapter has reviewed the research topics related to part build orientation determination for LPBF AM. Firstly, an overview of AM has been provided. This overview involves the definition, history and distinguishing characteristics of AM. Then, a brief introduction of the catego-

rised AM processes, including vat photopolymerisation, material jetting, binder jetting, powder bed fusion, material extrusion, directed energy deposition and sheet lamination, the LPBF process, and the AM product realisation process activities, including design, process planning, part build, post-processing and verification, has been made. After that, a review of the existing alternative orientation generation methods, including the exhaustive computation, feature recognition and facet clustering methods, and the existing optimal orientation selection methods, including the MOO and MADM methods, have been presented. The characteristics and limitations of each type of methods have been analysed in this review. Finally, the research gaps of the thesis have been identified.

3 The proposed automatic generation method

In this chapter, a novel facet clustering method for automatic generation of the ABOs of an LPBF part is presented. The schematic diagram of the presented method is shown in Figure 3.1. The method includes clustering of facets and generation of ABOs. In the clustering of facets, the STL model of an LPBF part, which is represented by a finite number of planar triangular facets, is used as the input of the accelerated HDBSCAN* algorithm (Campello et al., 2015; McInnes and Healy, 2017). A certain number of meaningful clusters of facets are produced via a specific clustering rule, the algorithm and the *k*-cluster lifetime partition criterion (Fred and Jain, 2005). In the generation of ABOs, whether the obtained facet clusters need to be further refined is first judged. If a refinement is no longer required, the ABOs of each cluster will be generated according to a specific generation rule and the ABOs of the part are obtained via combining the ABOs of all clusters. Otherwise, a refinement of clusters will be carried out according to a specific refinement rule and a smaller number of clusters will be achieved to generate the ABOs of the part.

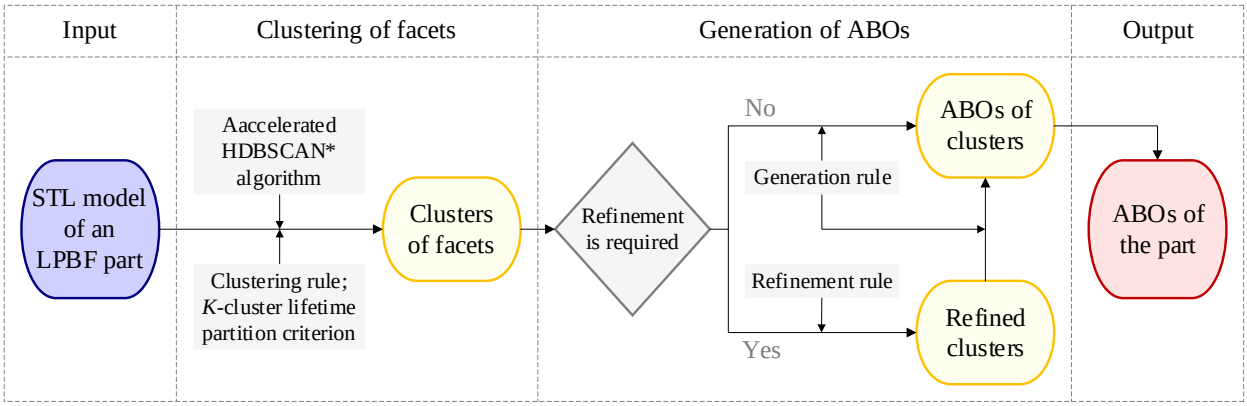


Figure 3.1 Schematic diagram of the automatic generation method

The chapter is organised as follows. Section 3.1 describes the specific process of the clustering of facets. The details of the generation of ABOs are explained in Section 3.2. Section 3.3 ends the chapter with a summary.

3.1 Clustering of planar triangular facets

There are many formats available for 3D model data representation in AM, such as the STL format, OBJ format, 3MF format and AMF format. Among the available formats, STL (Roscoe, 1988) is the most used one and has been the actual standard 3D model data format in AM. Almost all CAD systems can import and export STL files, and almost all AM machines include the

support of the STL format. Without loss of generality, the presented automatic generation method takes as input a 3D model encoded by the STL format.

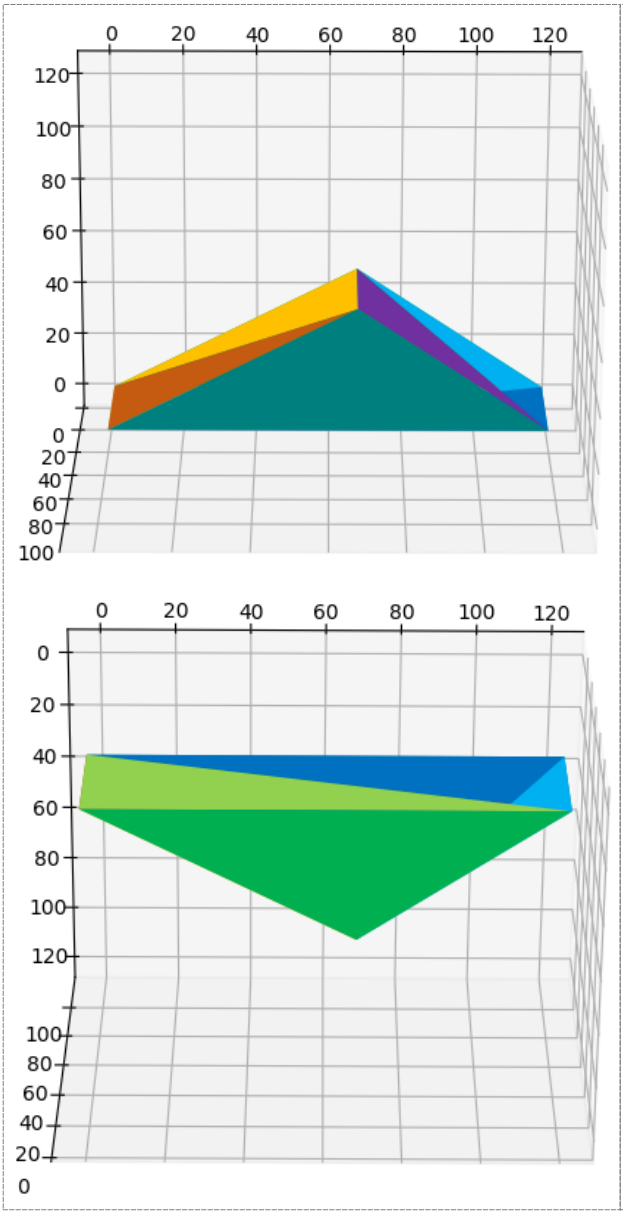
The STL format represents a 3D model via a technique called tessellation. It firstly uses the standard surface triangulation algorithm to triangulate the model. Then the surface of the model is covered by a set of adjoining planar triangular facets, where each facet is described by three vertices and one normal vector. Through encoding the vertices and normal vectors of all facets, an STL file of the 3D model is obtained. The STL format provides two ways to encode the vertices and normal vectors. One is ASCII (American Standard Code for Information Interchange) and the other is binary code. An ASCII STL file is both human readable and machine readable. It is mostly used for testing. An example is shown in Figure 3.2. As can be seen from Figure 3.2, the content of the ASCII STL file is very intuitive for reading. Compared to an ASCII STL file, a binary STL file is only machine readable and is mainly used for storage, as it needs less storage space than its ASCII version. As an example, the ASCII STL file of the prism in Figure 3.2 takes up 2.13 KB of storage space, while the size of its binary version is only 484 B. To ensure completeness, the presented method supports the input of both versions of STL files.

Taking an ASCII or a binary STL file as input, the presented method leverages the accelerated HDBSCAN* algorithm in (McInnes and Healy, 2017), which was developed by accelerating the original HDBSCAN* algorithm in (Campello et al., 2015), to automatically divide all planar triangular facets included in the STL file into a set of meaningful clusters. HDBSCAN* is an unsupervised learning algorithm for generating clusters of a given dataset. It starts with an assumption that there are some unknown probability density functions that can be used to draw the observed data objects. From a probability density function f defined on a metric space $(\boldsymbol{X}, d)$ (where $\boldsymbol{X}$ is a set of observed data objects and d is a distance metric), a hierarchical cluster structure in which a cluster is a connected subset of $\{x \mid x \in (\boldsymbol{X}, d) \wedge f(x) \leq \lambda\}$ can be constructed. This structure is called a dendrogram. Each cluster is one of its branches, which extends over the range of λ. The goal of the HDBSCAN* algorithm is to construct a dendrogram that can suitably approximate f in a hierarchical and nested way.

For the clustering of facets, the observed data objects are all planar triangular facets of an input STL model. The distance metric for facets can be customised according to specific AM processes, build materials and production requirements. From the studies of Wauthle et al. (2015), Brika et al. (2017), Cheng and To (2019) and Griffiths et al. (2019), the build orientation of an LPBF part could influence the support volume, volumetric error, surface roughness, build time, build cost, strength, elongation, hardness, residual stress, flexural modulus and fatigue perfor-

mance of the part. From this point of view, the distance metric can be selected to benefit some of these attributes. According to the study of Zhang et al. (2019), theoretically any two planar triangular facets that have similar normal vectors would have similar production results if they are built in the same orientation. Such similarity is measured via the angle between the normal vectors. The smaller the angle of two normal vectors, the more similar they are. Based on this theoretical conclusion, the presented method uses the following rule to perform facet clustering:

Rule 3.1 (Clustering rule). The observed data objects for the clustering of all planar triangular facets of an input STL model are the normal vectors of the facets and the distance metric is a metric of angle between the normal vectors of the facets in 3D space.




```
solid Prism
 0 facet normal -5.651917e-01 8.249596e-01 0.000000e+00
     outer loop
        vertex 0.000000e+00 0.000000e+00 4.000000e+01
        vertex 6.775755e+01 4.642168e+01 4.000000e+01
        vertex 0.000000e+00 0.000000e+00 0.000000e+00
     endloop
   endfacet
 1 facet normal -5.651917e-01 8.249596e-01 0.000000e+00
     outer loop
        vertex 6.775755e+01 4.642168e+01 4.000000e+01
        vertex 6.775755e+01 4.642168e+01 0.000000e+00
        vertex 0.000000e+00 0.000000e+00 0.000000e+00
     endloop
   endfacet
 2 facet normal 0.000000e+00 -1.000000e+00 0.000000e+00
     outer loop
        vertex 1.197395e+02 0.000000e+00 0.000000e+00
        vertex 0.000000e+00 0.000000e+00 4.000000e+01
        vertex 0.000000e+00 0.000000e+00 0.000000e+00
     endloop
   endfacet
 3 facet normal 0.000000e+00 -0.000000e+00 -1.000000e+00
     outer loop
        vertex 6.775755e+01 4.642168e+01 0.000000e+00
        vertex 1.197395e+02 0.000000e+00 0.000000e+00
        vertex 0.000000e+00 0.000000e+00 0.000000e+00
     endloop
   endfacet
 4 facet normal 6.660895e-01 7.458718e-01 0.000000e+00
     outer loop
        vertex 6.775755e+01 4.642168e+01 0.000000e+00
        vertex 1.197395e+02 0.000000e+00 4.000000e+01
        vertex 1.197395e+02 0.000000e+00 0.000000e+00
     endloop
   endfacet
 5 facet normal -0.000000e+00 -1.000000e+00 -0.000000e+00
     outer loop
        vertex 1.197395e+02 0.000000e+00 4.000000e+01
        vertex 0.000000e+00 0.000000e+00 4.000000e+01
        vertex 1.197395e+02 0.000000e+00 0.000000e+00
     endloop
   endfacet
 6 facet normal 0.000000e+00 0.000000e+00 1.000000e+00
     outer loop
        vertex 1.197395e+02 0.000000e+00 4.000000e+01
        vertex 6.775755e+01 4.642168e+01 4.000000e+01
        vertex 0.000000e+00 0.000000e+00 4.000000e+01
     endloop
   endfacet
 7 facet normal 6.660895e-01 7.458718e-01 -0.000000e+00
     outer loop
        vertex 6.775755e+01 4.642168e+01 0.000000e+00
        vertex 6.775755e+01 4.642168e+01 4.000000e+01
        vertex 1.197395e+02 0.000000e+00 4.000000e+01
     endloop
   endfacet
endsolid
```

Figure 3.2 The content of the ASCII STL file of a prism

Formally, suppose an STL model consists of n planar triangular facets and $\mathbf{n_0} = (x_0, y_0, z_0)$, $\mathbf{n_1} = (x_1, y_1, z_1)$, …, $\mathbf{n_{n-1}} = (x_{n-1}, y_{n-1}, z_{n-1})$ are their normal vectors. Then the observed data objects are $\mathbf{n}_0, \mathbf{n}_1, \ldots, \mathbf{n}_{n-1}$ and the distance metric is a metric of angle between $\mathbf{n_i} = (x_i, y_i, z_i)$ and $\mathbf{n_j} = (x_j, y_j, z_j)$ ($i, j = 0, 1, \ldots, n-1$ and $i \neq j$):

$$\theta(\mathbf{n_i}, \mathbf{n_j}) = \arccos \frac{x_i x_j + y_i y_j + z_i z_j}{\sqrt{x_i^2 + y_i^2 + z_i^2}\sqrt{x_j^2 + y_j^2 + z_j^2}} \tag{3.1}$$

It is found that using this metric can output clustering results in an acceptable time for regular surface models and is inefficient for freeform surface models, while using a metric of Euclidean distance between (x_i, y_i, z_i) and (x_j, y_j, z_j) can greatly improve the efficiency:

$$d(\mathbf{n_i}, \mathbf{n_j}) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{3.2}$$

The reason is that the amount of computation needed for Equation (3.1) is larger than the amount required for Equation (3.2). To ensure the efficiency of the presented method, $d(\mathbf{n_i}, \mathbf{n_j})$ is used as the distance metric of the accelerated HDBSCAN* algorithm. This is also effective for the generation of ABOs, because $d(\mathbf{n_i}, \mathbf{n_j})$ and $\theta(\mathbf{n_i}, \mathbf{n_j})$ are approximately linear when the angle is small:

$$\theta(\mathbf{n_i}, \mathbf{n_j}) = \arccos(1 - 0.5d^2(\mathbf{n_i}, \mathbf{n_j})) \tag{3.3}$$

This relationship is depicted in Figure 3.3. Based on this, the similarity between two normal vectors can also be measured by the Euclidean distance between their ends. When defining the distance metric of the accelerated HDBSCAN* algorithm, the distance can be set to 0.0001 for regular surface models (It should be set to 0, but the algorithm does not allow it to be 0) and transformed from angle according to Equation (3.3) for freeform surface models.

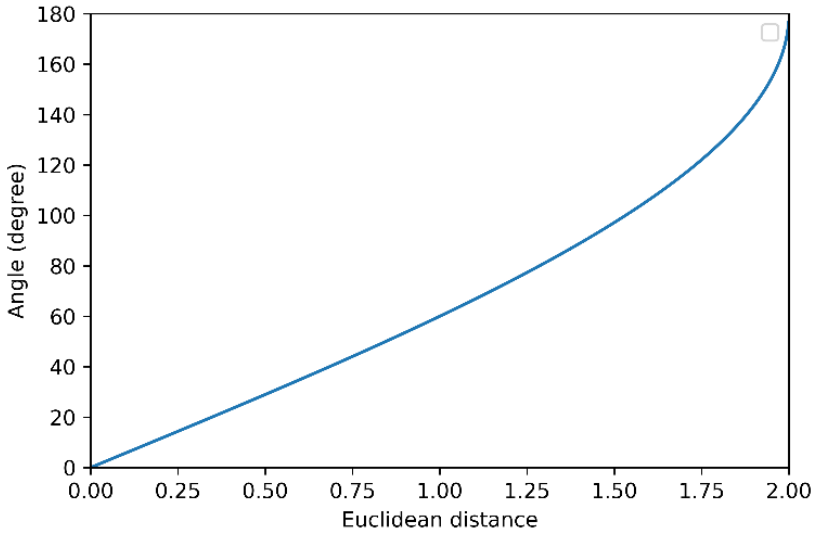


Figure 3.3 The relationship between angle and Euclidean distance

To suitably approximate the unknown probability density function of the normal vectors of n facets $\mathbf{n}_0$, $\mathbf{n}_1$, …, $\mathbf{n}_{n-1}$ in a hierarchical and nested way, the HDBSCAN* algorithm needs to perform six steps. Before describing these six steps, the following related concepts are first defined, which are derived from (McInnes and Healy, 2017).

Definition 3.1 (Core normal vector). A normal vector $\mathbf{n} \in \mathbf{N}$ (where $\mathbf{N} = \{\mathbf{n}_0, \mathbf{n}_1, \ldots, \mathbf{n}_{n-1}\}$) is a core normal vector with respect to a distance scale ε and the minimum number of normal vectors in a neighbourhood for a normal vector to be considered as a core normal vector m if the ε-neighbourhood of $\mathbf{n}$ contains at least m normal vectors, i.e. $|\{\mathbf{n}' \in \mathbf{N} \mid d(\mathbf{n}', \mathbf{n}) \leq \varepsilon\}| \geq m$ (where $|\cdot|$ denotes the cardinality of a set).

Definition 3.2 (ε-reachable). Let $\mathbf{n}_i$ and $\mathbf{n}_j$ be two core normal vectors. They are ε-reachable with respect to ε and m if $\mathbf{n}_i \in \{\mathbf{n}' \in \mathbf{N} \mid d(\mathbf{n}', \mathbf{n}_j) \leq \varepsilon\}$ and $\mathbf{n}_j \in \{\mathbf{n}' \in \mathbf{N} \mid d(\mathbf{n}', \mathbf{n}_i) \leq \varepsilon\}$.

Definition 3.3 (Density-connected). Let $\mathbf{n}_i$ and $\mathbf{n}_j$ be two core normal vectors. They are density-connected with respect to ε and m if they are ε-reachable.

Definition 3.4 (Cluster of normal vectors). Let $\boldsymbol{C}$ be a non-empty maximal subset of $\mathbf{N}$. Then $\boldsymbol{C}$ is a cluster of normal vectors with respect to ε and m if each pair of normal vectors in $\boldsymbol{C}$ is density-connected.

Definition 3.5 (Core distance). Let $\mathbf{n}$ be a normal vector in $\mathbf{N}$. The core distance of $\mathbf{n}$ with respect to m, denoted as $d_c(\mathbf{n})$, is the distance between the end of $\mathbf{n}$ and the end of its m-th nearest neighbour (including $\mathbf{n}$ itself).

Definition 3.6 (Mutual reachability distance). Let $\mathbf{n}_i$ and $\mathbf{n}_j$ be two normal vectors in $\mathbf{N}$. The mutual reachability distance between them with respect to m is $d_{mr}(v) = \max\{d_c(\mathbf{n}_i), d_c(\mathbf{n}_j), d(\mathbf{n}_i, \mathbf{n}_j)\}$ if $\mathbf{n}_i \neq \mathbf{n}_j$ and $d_{mr}(\mathbf{n}) = 0$ if $\mathbf{n}_i = \mathbf{n}_j$.

Definition 3.7 (Mutual reachability graph). The mutual reachability graph of $\mathbf{N}$ is a complete graph $\boldsymbol{G}_{mr}$, where the normal vectors in $\mathbf{N}$ are vertices and the mutual reachability distance of its two vertices is the weight of each edge.

Definition 3.8 (Minimum spanning tree). Let $\boldsymbol{G}$ be a connected weighted undirected graph. The minimum spanning tree of $\boldsymbol{G}$ is a subset of the edges of $\boldsymbol{G}$ that connects all vertices without any cycles and with the minimum total edge weight.

Based on the definitions above, the six steps performed in the HDBSCAN* algorithm are described as follows:

- Calculate the core distances of $\mathbf{n}_0$, $\mathbf{n}_1$, …, $\mathbf{n}_{n-1}$ with respect to m. The minimum number of normal vectors in a neighbourhood for a normal vector to be considered a core normal vector for clustering of facets is one, thus m is assigned 1.

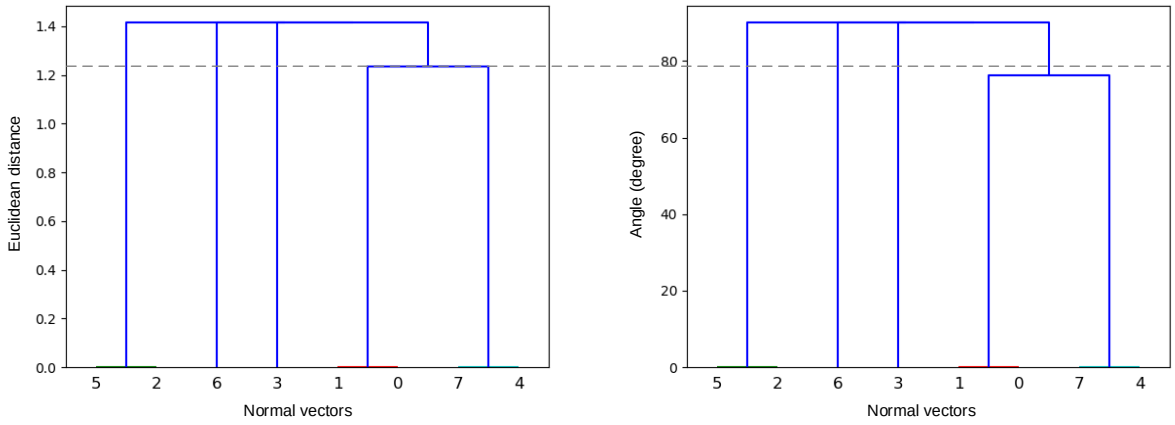
- Compute the mutual reachability distance of each pair of $\mathbf{n}_i$ and $\mathbf{n}_j$ in $\mathbf{N}$ with respect to m.
- Construct a mutual reachability graph of $\mathbf{n}$.
- Produce a minimum spanning tree of the constructed graph.
- Extend the minimum spanning tree via adding a self edge for each vertex. The weight of the added edge is set as the core distance of the corresponding normal vector.
- Construct a dendrogram according to the extended minimum spanning tree. All normal vectors are assigned the same label for the root of the extended minimum spanning tree. All edges from the extended minimum spanning tree are iteratively removed in descending order of the weights. The weight of the edge to be removed is assigned to ε before each removal. The connected components which contain the end vertex of the removed edge are assigned labels to achieve the next hierarchical level after each removal. If a connected component still has at least one edge, then a new label is assigned to it. Otherwise, a null label is assigned to the connected component.

As can be obtained from (McInnes and Healy, 2017), the HDBSCAN* algorithm on n planar triangular facets (i.e. n normal vectors) has $O(n^2)$ run-time. This complexity is slightly high for the clustering of facets, since the number of facets for complex STL models, especially freeform surface STL models, is usually very large (e.g. 500,000, 1,000,000). To improve the efficiency of the HDBSCAN* algorithm, space tree algorithms (Ram et al., 2009) and Boruvka's algorithm (Nesetril et al., 2001) were respectively leveraged to accelerate the calculation of distances and the production of a minimum spanning tree in (Campello et al., 2015). These accelerations derived an accelerated HDBSCAN* algorithm, which reduces the time complexity to $O(n\log n)$. To this end, the presented method also uses these algorithms to improve the efficiency of the clustering of planar triangular facets.

The output of the accelerated HDBSCAN* algorithm for a specific STL model is a dendrogram that depicts the clustering process of the normal vectors of all its planar triangular facets. On the basis of this dendrogram, the presented method uses the k-cluster lifetime partition criterion (Fred and Jain, 2005) to identify k meaningful clusters of facets. The identification process includes the following steps:

- Convert the metric of the ordinate of the dendrogram from Euclidean distance to angle using Equation (3.3) to establish a new dendrogram whose abscissa denotes normal vectors and ordinate denotes angle. Though the Euclidean distance between two normal vectors and the angle between them have the same monotonicity, they are not completely linear, which means that the dendrogram constructed by the accelerated HDBSCAN* algorithm cannot be directly

used in the later computation. Before the computation, Euclidean distance is converted to angle using Equation (3.3). As an example, a dendrogram constructed by the algorithm for the prism in Figure 3.2 is shown in Figure 3.4(a). It is reconstructed in Figure 3.4(b) via converting Euclidean distance to angle.



(a) The constructed dendrogram for the prism in Figure 3.2

(b) The constructed new dendrogram after conversion

Figure 3.4 The dendrogram for the prism and its converted dendrogram

- Compute *k*-cluster lifetimes for $k = 1, 2, \ldots, n$. Lifetime is originally defined as the distance between that a cluster is created and that it merges with other clusters during clustering. For instance, the *k*-cluster lifetimes ($k = 1, 2, \ldots, 8$) for the dendrogram in Figure 3.4(b) are computed as follows:

  1-cluster lifetime = 00.00° − 00.00° = 00.00°

  2-cluster lifetime = 00.00° − 00.00° = 00.00°

  3-cluster lifetime = 00.00° − 00.00° = 00.00°

  4-cluster lifetime = 90.00° − 76.18° = 13.82°

  5-cluster lifetime = 76.18° − 00.00° = 76.18°

  6-cluster lifetime = 00.00° − 00.00° = 00.00°

  7-cluster lifetime = 00.00° − 00.00° = 00.00°

  8-cluster lifetime = 00.00° − 00.00° = 00.00°

- Find the largest lifetime from all *k*-cluster lifetimes and output its corresponding clusters. For example, the largest lifetime in the dendrogram in Figure 3.4(b) is 5-cluster lifetime. The five clusters are $\boldsymbol{C}_1 = \{\mathbf{n}_5, \mathbf{n}_2\}$, $\boldsymbol{C}_2 = \{\mathbf{n}_6\}$, $\boldsymbol{C}_3 = \{\mathbf{n}_3\}$, $\boldsymbol{C}_4 = \{\mathbf{n}_1, \mathbf{n}_0\}$ and $\boldsymbol{C}_5 = \{\mathbf{n}_7, \mathbf{n}_4\}$. If the planar triangular facets in the same cluster are painted the same colour, then the clustering result depicted in Figure 3.5 will be obtained.

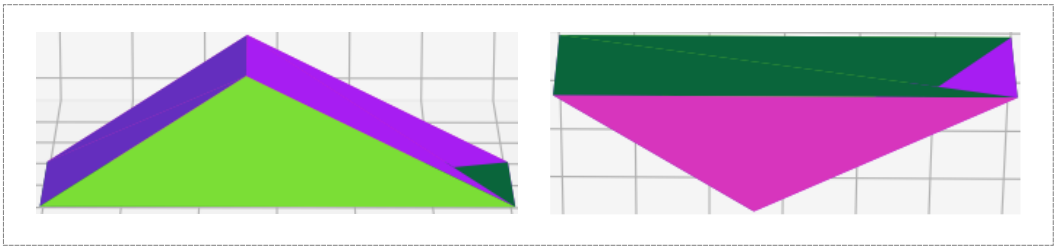


Figure 3.5 The clustering result of the facets of the prism

3.2 Generation of alternative orientations

The generation of ABOs starts from judging whether the identified clusters require further refinement. This is because the number of the produced clusters is usually very large for an STL model with complex shape or freeform surfaces; and a large number of clusters will bring a lot of meaningless computations to further OBO selection. The basis for the judgment is the number of the output clusters together with the type of the input STL model. In general, the number of the output clusters should be within 6 (12) when the STL model is a regular (freeform) surface model consisting of less than 120,000 facets. A further refinement is required if such conditions are not satisfied. In some cases, the judgment can also be carried out according to the production experience of a user. For example, the produced clusters in Figure 3.5 no longer require further refinement, because the STL model of the prism is a regular surface model and the number of the output clusters is 5. However, a further refinement is required for the output clusters of a regular surface model consisting of 276 facets in Figure 3.6 and is also required for the output clusters of a freeform surface model consisting of 13,240 facets in Figure 3.7, as the number of the output clusters for the former STL model (95) has exceeded 6 and this number for the latter STL model (6,670) has exceeded 12.

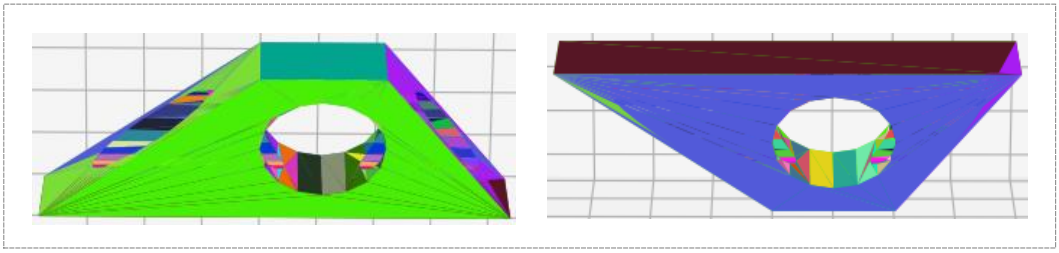


Figure 3.6 The clustering result of the facets of a regular surface model

The produced facet clusters will be leveraged directly to generate meaningful ABOs if a further refinement is not required. To generate meaningful ABOs for each facet cluster, a computa-

tional method or a rule-based method can be applied, which is similar to the feature recognition methods for generation of ABOs (Cheng et al., 1995; Frank and Fadel, 1995; Lan et al., 1997; Alexander et al., 1998; Pham et al., 1999; Xu et al., 1999; West et al., 2001; Byun and Lee, 2006; Zhang et al., 2016; Al-Ahmari et al., 2018). The presented method adopts a central normal vector-based generation rule developed in the facet clustering method of Zhang et al. (2019):

Rule 3.2 (Generation rule). The unitised central vector of all normal vectors in a cluster and its opposite vector directly serve as the ABOs of this cluster.

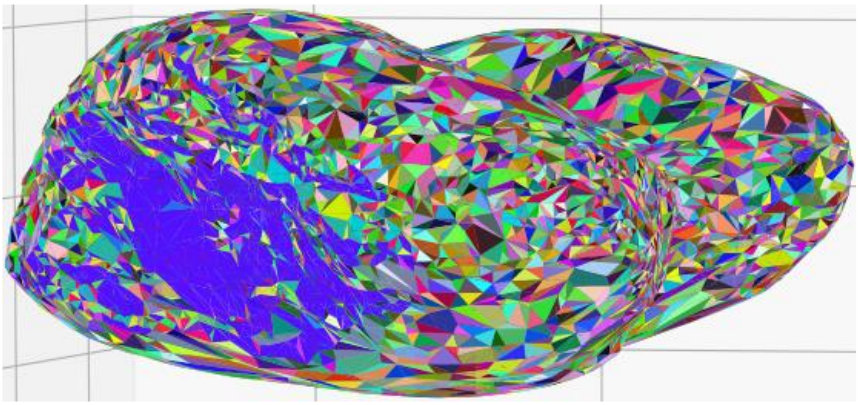


Figure 3.7 The clustering result of the facets of a freeform surface model

Using this rule, the ABOs of each cluster of facets of the STL model of an LPBF part can be computed. The ABOs of the part are obtained via combining the computed ABOs of all clusters and removing duplicated ABOs among them.

Formally, let $\boldsymbol{C_i} = \{\mathbf{n}_{i,0}, \mathbf{n}_{i,1}, \ldots, \mathbf{n}_{i,n_i-1}\}$ (where $\mathbf{n}_{i,0} = (x_{i,0}, y_{i,0}, z_{i,0})$, $\mathbf{n}_{i,1} = (x_{i,1}, y_{i,1}, z_{i,1})$, …, $\mathbf{n}_{i,n_i-1} = (x_{i,n_i-1}, y_{i,n_i-1}, z_{i,n_i-1})$) be an arbitrary cluster of normal vectors of facets. Then the unitised central vector of all normal vectors in this cluster is as follow:

$$\mathbf{n}_{\mathrm{c},i} = \frac{\left(\sum_{j=0}^{n_i-1} x_{i,j}, \sum_{j=0}^{n_i-1} y_{i,j}, \sum_{j=0}^{n_i-1} z_{i,j} \right)}{\sqrt{\left(\sum_{j=0}^{n_i-1} x_{i,j} \right)^2 + \left(\sum_{j=0}^{n_i-1} y_{i,j} \right)^2 + \left(\sum_{j=0}^{n_i-1} z_{i,j} \right)^2}} \tag{3.4}$$

The opposite vector of this unit vector is $-\mathbf{n}_{\mathrm{c},i}$. The unit vectors $\pm\mathbf{n}_{\mathrm{c},i}$ are taken as the ABOs of the cluster $\boldsymbol{C_i}$ based on Rule 3.2. Suppose the facets of the STL model of an LPBF part are partitioned into N clusters $\boldsymbol{C_1}, \boldsymbol{C_2}, \ldots, \boldsymbol{C_N}$ and $\pm\mathbf{n}_{\mathrm{c},1}, \pm\mathbf{n}_{\mathrm{c},2}, \ldots, \pm\mathbf{n}_{\mathrm{c},N}$ are respectively the ABOs of the N clusters, the ABOs of the part are generated as $\pm\mathbf{n}_{\mathrm{c},1}, \pm\mathbf{n}_{\mathrm{c},2}, \ldots, \pm\mathbf{n}_{\mathrm{c},N}$, in which the same vectors

are removed. For instance, the ABOs of the five clusters $\boldsymbol{C}_1 = \{\mathbf{n}_5, \mathbf{n}_2\}$, $\boldsymbol{C}_2 = \{\mathbf{n}_6\}$, $\boldsymbol{C}_3 = \{\mathbf{n}_3\}$, $\boldsymbol{C}_4 = \{\mathbf{n}_1, \mathbf{n}_0\}$ and $\boldsymbol{C}_5 = \{\mathbf{n}_7, \mathbf{n}_4\}$ depicted in Figure 3.5 are respectively as follows:

$+\mathbf{n}_{c,1}$ = (00.0000, −1.0000, 00.0000)

$-\mathbf{n}_{c,1}$ = (00.0000, +1.0000, 00.0000)

$+\mathbf{n}_{c,2}$ = (−0.5652, +0.8250, 00.0000)

$-\mathbf{n}_{c,2}$ = (+0.5652, −0.8250, 00.0000)

$+\mathbf{n}_{c,3}$ = (+0.6661, +0.7459, 00.0000)

$-\mathbf{n}_{c,3}$ = (−0.6661, −0.7459, 00.0000)

$+\mathbf{n}_{c,4}$ = (00.0000, 00.0000, +1.0000)

$-\mathbf{n}_{c,4}$ = (00.0000, 00.0000, −1.0000)

$+\mathbf{n}_{c,5}$ = (00.0000, 00.0000, −1.0000)

$-\mathbf{n}_{c,5}$ = (00.0000, 00.0000, +1.0000)

The unit vectors $+\mathbf{n}_{c,5}$ and $-\mathbf{n}_{c,5}$ are removed, because they are respectively the same as the unit vectors $-\mathbf{n}_{c,4}$ and $+\mathbf{n}_{c,4}$. Therefore, the ABOs of the prism in Figure 3.2 are generated as follows:

O_1 = (00.0000, −1.0000, 00.0000)

O_2 = (00.0000, +1.0000, 00.0000)

O_3 = (−0.5652, +0.8250, 00.0000)

O_4 = (+0.5652, −0.8250, 00.0000)

O_5 = (+0.6661, +0.7459, 00.0000)

O_6 = (−0.6661, −0.7459, 00.0000)

O_7 = (00.0000, 00.0000, +1.0000)

O_8 = (00.0000, 00.0000, −1.0000)

The schematic diagram of these ABOs is shown in Figure 3.8.

The produced facet clusters will be refined to screen out a smaller number of facet clusters to generate meaningful ABOs if a further refinement is needed. To carry out the refinement, customised rules can be applied. A refinement rule aiming to select a certain number of facet clusters with relatively large areas can be used. This is equivalent to find the base planes of an STL model (Lan et al., 1997). The refinement rule is described as follow:

Rule 3.3 (Refinement rule). The top 6 (12) facet clusters in area are used to generate ABOs if the input STL model is a regular (freeform) surface model consisting of less than 120,000 facets.

Using this rule, 6 or 12 facet clusters are screened out, which can be used to generate ABOs. It is worth noting that 6 and 12 in Rule 3.3 are two experimental numbers. They are respectively obtained through carrying out orientation experiments on thirty regular surface models and thirty

freeform surface models, each of which consists of less than 120,000 facets. If a model contains more than 120,000 facets, users can adjust the two numbers if the OBO of the model is not included in the generated ABOs. It should be noted that the smaller the number of the facet clusters used to generate ABOs, the less computation required in selection of the OBO.

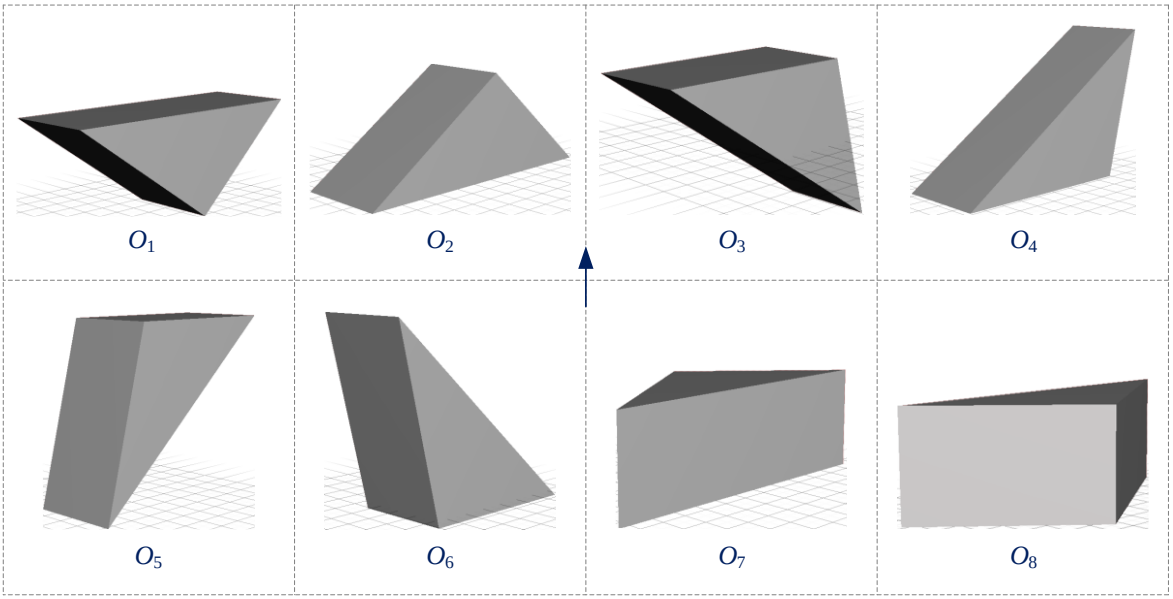


Figure 3.8 Schematic diagram of the ABOs of the prism

Formally, let $\boldsymbol{C_i} = \{\mathbf{n}_{i,0}, \mathbf{n}_{i,1}, \ldots, \mathbf{n}_{i,n_i-1}\}$ be an arbitrary cluster of normal vectors of facets and $F_{i,0} = (v_{i,0,1}, v_{i,0,2}, v_{i,0,3})$, $F_{i,1} = (v_{i,1,1}, v_{i,1,2}, v_{i,1,3})$, …, $F_{i,n_i-1} = (v_{i,n_i-1,1}, v_{i,n_i-1,2}, v_{i,n_i-1,3})$ be respectively the planar triangular facets corresponding to $\mathbf{n}_{i,0}, \mathbf{n}_{i,1}, \ldots, \mathbf{n}_{i,n_i-1}$. The area of $\boldsymbol{C_i}$ can be computed using the following equation:

$$\boldsymbol{A}(\boldsymbol{C}_i) = \sum_{j=0}^{n_i-1} \sqrt{p_{i,j}\left(p_{i,j} - d(v_{i,j,1}, v_{i,j,2})\right)\left(p_{i,j} - d(v_{i,j,1}, v_{i,j,3})\right)\left(p_{i,j} - d(v_{i,j,2}, v_{i,j,3})\right)} \tag{3.5}$$

where $d(v_{i,j,1}, v_{i,j,2})$, $d(v_{i,j,1}, v_{i,j,3})$ and $d(v_{i,j,2}, v_{i,j,3})$ are respectively the lengths of the three edges of the planar triangular facet $F_{i,j}$ that can be computed using Equation (3.2), and

$$p_{i,j} = \frac{1}{2}\left(d(v_{i,j,1}, v_{i,j,2}) + d(v_{i,j,1}, v_{i,j,3}) + d(v_{i,j,2}, v_{i,j,3})\right) \tag{3.6}$$

Assume the facets of the STL model of an LPBF part are divided into N clusters $\boldsymbol{C_1}, \boldsymbol{C_2}, \ldots, \boldsymbol{C_N}$. Then the area of $\boldsymbol{C_i}$ ($i = 1, 2, \ldots, N$) is calculated using Equation (3.5) and the top 6 (12) clusters in area are screened out. The ABOs of the screened out clusters are calculated according to Rule 3.2. The ABOs of the part are obtained via gathering the calculated ABOs of all screened out clusters and removing duplicated ABOs among them.

As an example, the produced clusters in Figure 3.6 are refined using Rule 3.3 and the top 6 clusters in area, as depicted in Figure 3.9 (Please note that the remaining facets are displayed in white), are screened out. According to Rule 3.2, eight ABOs are generated as the ABOs of the corresponding LPBF part. The OBO of the part can be selected from these ABOs. Table 3.1 provides the unitised vectors of the generated ABOs. The schematic diagram of the ABOs is shown in Figure 3.10.

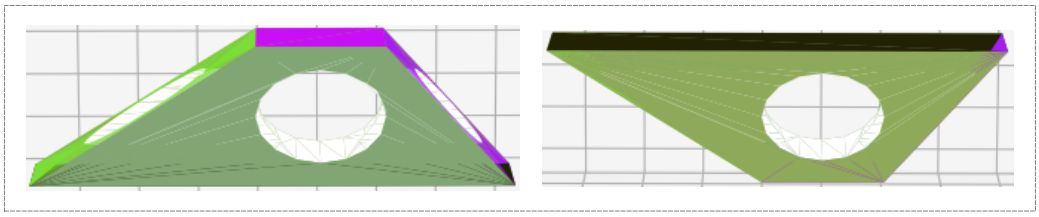


Figure 3.9 The top six clusters in area of the regular surface model

Table 3.1 The unitised vectors of the ABOs of the regular surface model

ABO	Unitised vector	ABO	Unitised vector
O_1	(00.0000, −1.0000, 00.0000)	O_5	(−0.6508, +0.7593, 00.0000)
O_2	(00.0000, +1.0000, 00.0000)	O_6	(+0.6508, −0.7593, 00.0000)
O_3	(00.0000, 00.0000, −1.0000)	O_7	(+0.8321, +0.5547, 00.0000)
O_4	(00.0000, 00.0000, +1.0000)	O_8	(−0.8321, −0.5547, 00.0000)

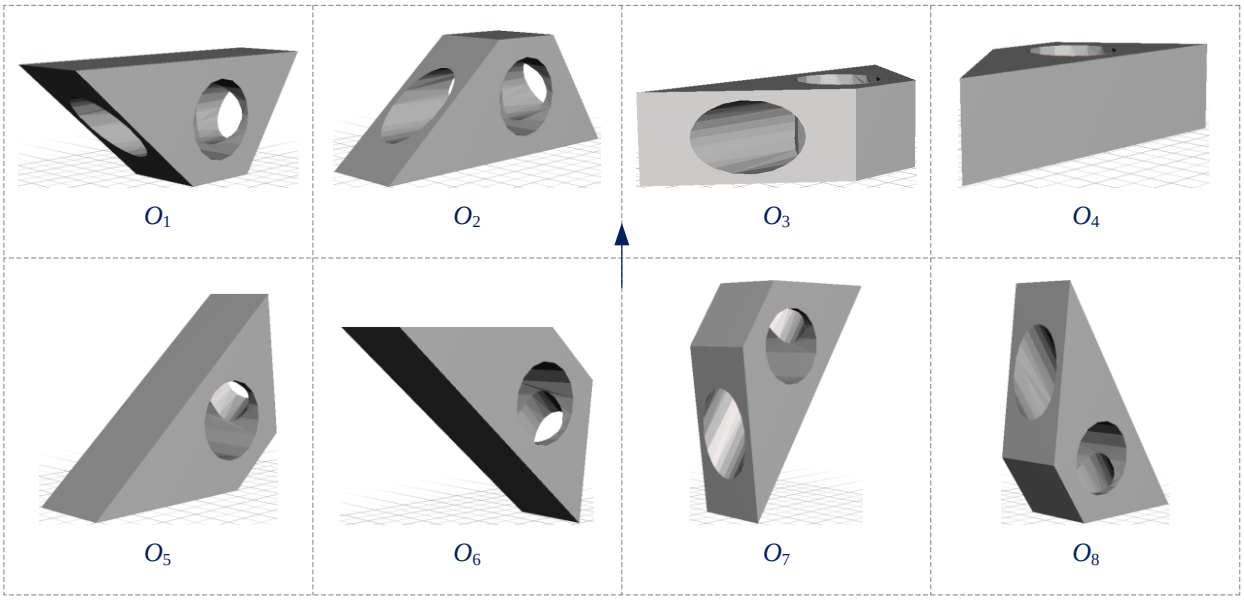


Figure 3.10 Schematic diagram of the ABOs of the regular surface model

Similarly, the produced clusters in Figure 3.7 are refined using Rule 3.3 and the top 12 clusters in area, as depicted in Figure 3.11 (Please note that the remaining facets are also displayed in

white), are screened out. According to Rule 3.2, twenty-four ABOs are generated as the ABOs of the corresponding LPBF part. The OBO of the part can be selected from these ABOs. Table 3.2 lists the unitised vectors of the generated ABOs. The schematic diagram of the ABOs is provided in Figure 3.12.

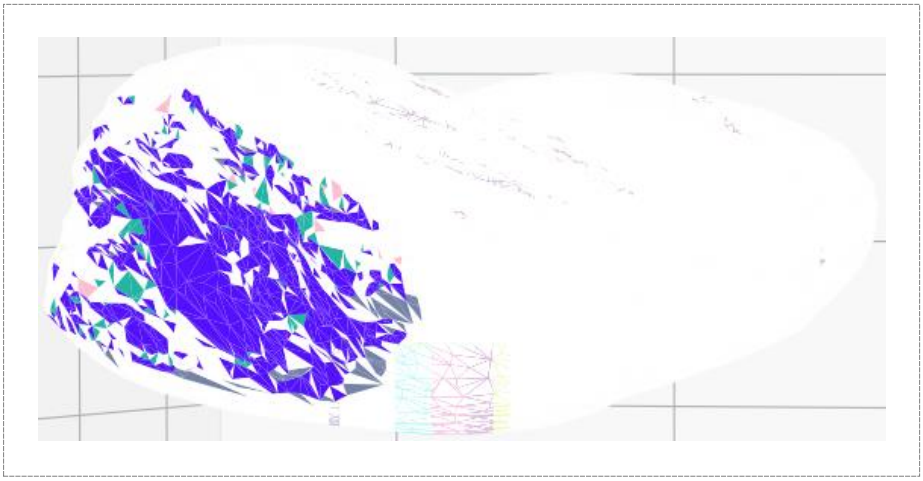


Figure 3.11 The top twelve clusters in area of the freeform surface model

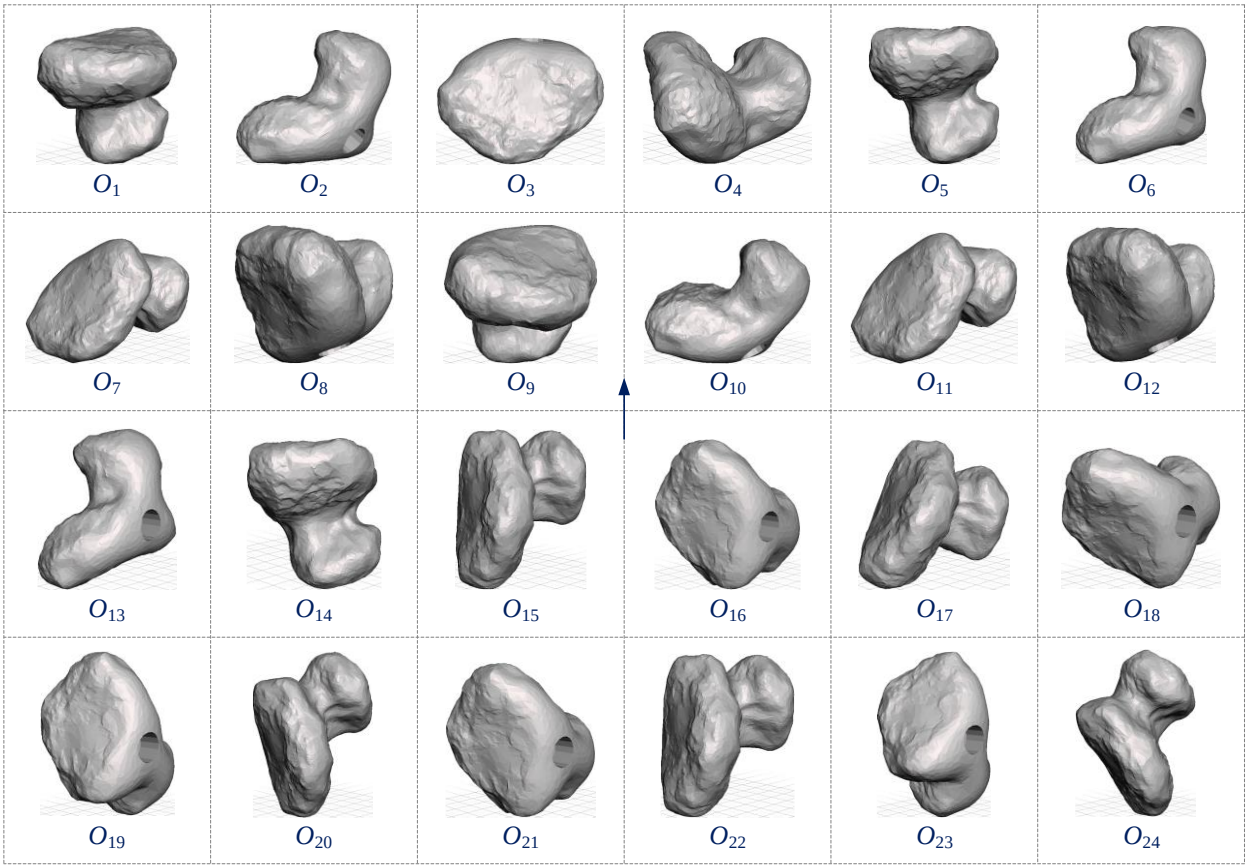


Figure 3.12 Schematic diagram of the ABOs of the freeform surface model

Table 3.2 The unitised vectors of the ABOs of the freeform surface model

ABO	Unitised vector	ABO	Unitised vector	ABO	Unitised vector
O_1	(−0.6801, +0.6591, −0.3210)	O_9	(−0.6713, +0.4947, −0.5520)	O_{17}	(+0.1963, +0.9805, −0.0015)
O_2	(+0.6801, −0.6591, +0.3210)	O_{10}	(+0.6713, −0.4947, +0.5520)	O_{18}	(−0.1963, −0.9805, +0.0015)
O_3	(+0.0007, −0.0001, −0.9999)	O_{11}	(+0.4607, −0.5408, +0.7038)	O_{19}	(+0.8314, +0.5557, 00.0000)
O_4	(−0.0007, +0.0001, +0.9999)	O_{12}	(−0.4607, +0.5408, −0.7038)	O_{20}	(−0.8314, −0.5557, 00.0000)
O_5	(−0.7549, +0.6460, −0.1133)	O_{13}	(−0.8315, +0.5555, +0.0046)	O_{21}	(+0.5589, −0.8293, −0.0001)
O_6	(+0.7549, −0.6460, +0.1133)	O_{14}	(+0.8315, −0.5555, −0.0046)	O_{22}	(−0.5589, +0.8293, +0.0001)
O_7	(−0.4238, +0.4943, −0.7590)	O_{15}	(+0.5592, +0.8290, −0.0052)	O_{23}	(+0.9807, +0.1955, 00.0000)
O_8	(−0.4238, +0.4943, −0.7590)	O_{16}	(−0.5592, −0.8290, +0.0052)	O_{24}	(−0.9807, −0.1955, 00.0000)

3.3 Summary

This chapter has presented a facet clustering based method for automatic generation of ABOs for LPBF AM. This method includes clustering of facets and generation of ABOs. The clustering of facets takes the STL model of an LPBF part as input and adopts the accelerated HDBSCAN* algorithm and the *k*-cluster lifetime partition criterion to divide all facets of the STL model into a certain number of meaningful clusters, each of which shares a similar normal vector. The generation of ABOs directly generates the ABOs of the part using a generation rule if the produced facet clusters are not needed to be refined, or firstly refines the facet clusters using a refinement rule and then generates the ABOs via the generation rule when a further refinement is required. The presented facet clustering method will be demonstrated in Chapter 6.

4 The constructed fuzzy aggregation operators

MADM refers to the process of finding desirable alternatives via synthetically evaluating the values of multiple attributes of all alternatives. There are two critical tasks in this process. One is to quantify the values of different attributes; and the other is to synthetically assess the quantified attributes to determine desirable alternatives (Greco et al., 2016). For the quantification of values of attributes, an important mathematical tool is the fuzzy number (Zadeh, 1965). Fuzzy number is the core component of a fuzzy set. The value of a fuzzy number is restricted to [0, 1] and can be used to quantify the degree of membership of an element to the fuzzy set. This can provide a normalisation of the attribute values in MADM to make them easy to process. Because of such capability, fuzzy number and its derived research topics have received extensive attention in the field of MADM. A large number of fuzzy MADM methods have been presented during the past few decades (Kahraman, 2008).

For the assessment of the quantified attributes, there are generally two approaches. One is to apply conventional MADM methods and the other is to use AOs. An AO is a mathematical function in which multiple values are grouped together to obtain a single summary value (Grabisch et al., 2009). AOs-based approaches are significantly more convincing and suitable than traditional MADM methods, because they can provide summary values of attributes and rankings of alternatives while traditional MADM methods only produce rankings (Liu and Wang, 2019). So far, there have been a number of AOs for MADM (Mardani et al., 2018). The MM and GMM operators (Muirhead, 1902) and the PA operator (Yager, 2001) are three of them. The MM and GMM operators can capture the interrelationships of the aggregated arguments. They are found to generate opposite aggregation expectations in MADM. An average combination of their aggregation results can balance the opposite aggregation expectations. The PA operator is capable of reducing the negative influence of biased argument values on the aggregation result.

In practical OBO selection, the considered attributes of ABOs are generally not independent of each other, but are interrelated. An ideal MADM method for OBO selection should be general and flexible enough to capture the interrelationships of the considered attributes to obtain more reasonable results. In addition, the values of attributes are generally obtained through theoretical calculation, simulation estimation or expert evaluation. It is usually difficult to ensure the absolute objectivity of these ways, which means that there would be some biased attribute values. To obtain reasonable selection results in this situation, it is of great necessity to reduce the negative effect of unreasonable attribute values.

On the basis of the considerations above, the objectives of the present chapter are outlined as follows:

- To make the values of attributes of ABOs easier to process, fuzzy numbers are introduced to normalise these values.
- To develop an AOs-based MADM method for OBO selection which can capture the complex relationships of attributes of ABOs, the MM and GMM operators of fuzzy numbers are used as the core AOs of the method.
- To make the method capable to reduce the negative impact of extreme attribute values on the aggregation result, the PA operator of fuzzy numbers is combined with the MM and GMM operators of fuzzy numbers.
- To express the relative importance of attributes of ABOs, weights are combined with the PA, MM and GMM operators of fuzzy numbers.
- To perform the operations in the combined operators, a set of operational laws of fuzzy numbers based on the Archimedean t-norm and t-conorm are applied.

To sum up, the present chapter aims to construct a set of fuzzy Archimedean weighted power MM and GMM operators for aggregating the values of attributes of ABOs. This aim is achieved through combining the MM and GMM operators of fuzzy numbers with the PA operator of fuzzy numbers, weights and the operational laws of fuzzy numbers based on the Archimedean t-norm and t-conorm. Because of such combination, the constructed fuzzy AOs will combine all of their capabilities. In Chapter 5, the constructed fuzzy AOs will be applied to develop a novel MADM method for automatic selection of the OBO of an LPBF part.

The remainder of this chapter is organised as follows. Section 4.1 introduces some prerequisites. A fuzzy Archimedean weighted power Muirhead mean (FAWPMM) operator and a fuzzy Archimedean weighted power geometric Muirhead mean (FAWPGMM) operator are constructed in Section 4.2 and Section 4.3, respectively. Section 4.4 ends the chapter with a summary.

4.1 Preliminaries

In this section, some prerequisites related to the fuzzy set theory, MM operator, GMM operator, PA operator, Archimedean t-norm and t-conorm and operational laws of fuzzy numbers are briefly introduced.

4.1.1 Fuzzy set theory

The concept of fuzzy set was introduced by Zadeh (1965) as an extension of the classical set. Its formal definition is as follow:

Definition 4.1 (Fuzzy set). A fuzzy set A in a finite universe of discourse X is $\{<x, \mu_A(x)> \mid x \in X\}$, where $\mu_A : X \rightarrow [0, 1]$ denotes the degree of membership of the element $x \in X$ to the set A, with the condition that $0 \leq \mu_A(x) \leq 1$.

Generally, the values of the membership function $\mu_A(x)$ are called as fuzzy numbers. The definition of a fuzzy number can be naturally obtained as follow (Zadeh, 1965):

Definition 4.2 (Fuzzy number). A fuzzy number α on a fuzzy set $A = \{<x, \mu_A(x)> \mid x \in X\}$ is $<\mu_A(x)>$. For the sake of simplicity, the fuzzy number α is always denoted as $<\mu>$.

To compare two fuzzy numbers, their score values are required. The following is the formal definition of the score value of a fuzzy number (Klir and Yuan, 1995):

Definition 4.3 (Score value). Let $\alpha = <\mu>$ be a fuzzy number. Its score value can be calculated using the following equation:

$$S(\alpha) = 2\mu - 1 \tag{4.1}$$

Any two fuzzy numbers can be compared via comparing their score values. The following is the formal definition of the rule for such comparison (Klir and Yuan, 1995):

Definition 4.4 (Comparison rule). Let $\alpha_1 = <\mu_1>$ and $\alpha_2 = <\mu_2>$ be two fuzzy numbers and $S(\alpha_1)$ and $S(\alpha_2)$ be respectively their score values. Then: If $S(\alpha_1) < S(\alpha_2)$, then $\alpha_1 < \alpha_2$; If $S(\alpha_1) = S(\alpha_2)$, then $\alpha_1 = \alpha_2$; If $S(\alpha_1) > S(\alpha_2)$, then $\alpha_1 > \alpha_2$.

To compute the distance between two fuzzy numbers, a distance measure of fuzzy numbers is required. The following is the formal definition of the Euclidean distance between two fuzzy numbers:

Definition 4.5 (Euclidean distance). Let $\alpha_1 = <\mu_1>$ and $\alpha_2 = <\mu_2>$ be two fuzzy numbers. The Euclidean distance between them can be computed using the following equation:

$$d(\alpha_1, \alpha_2) = |\mu_1 - \mu_2| \tag{4.2}$$

4.1.2 Muirhead mean (MM) operator

The MM operator was introduced by Muirhead (1902). This operator is capable of capturing the interrelationships of multiple aggregated arguments and can provide a general form of the arithmetic average (AA) operator, Bonferroni mean (BM) operator, Maclaurin symmetric mean (MSM) operator and geometric average (GA) operator. Its formal definition is as follow:

Definition 4.6 (MM operator). Let $a_1, a_2, \ldots, a_n$ be n positive numbers, $Q = (q_1, q_2, \ldots, q_n)$ be a collection of n real numbers such that $q_1, q_2, \ldots, q_n \geq 0$ but not at the same time $q_1 = q_2 = \ldots = q_n = 0$, $p(i)$ be a permutation of $(1, 2,\ldots, n)$ and $\boldsymbol{P}_n$ be the set of all permutations of $(1, 2,\ldots, n)$. Then the aggregation function

$$MM^{Q}(a_1, a_2, ..., a_n) = \left(\frac{1}{n!} \sum_{p \in \boldsymbol{P}_n} \prod_{i=1}^{n} a_{p(i)}^{q_i} \right)^{\frac{1}{\sum_{i=1}^{n} q_i}} \tag{4.3}$$

is called the MM operator.

In the MM operator, whether the interrelationships of a_1, a_2, …, a_n are captured relies on the values of q_1, q_2, …, q_n: If $q_1 > 0$ and $q_2 = q_3 = \ldots = q_n = 0$, then the interrelationships are not captured; If q_1, $q_2 > 0$ and $q_3 = q_4 = \ldots = q_n = 0$, then the interrelationships between two of a_1, a_2, …, a_n are captured; If q_1, q_2, …, $q_k > 0$ ($k = 3, 4, \ldots, n$) and $q_{k+1} = q_{k+2} = \ldots = q_n = 0$, then the interrelationships among k of a_1, a_2, …, a_n are captured.

In addition, the MM operator will reduce to the AA, BM, MSM and GA operators when q_1, q_2, …, q_n take some special values:

- The MM operator will reduce to the AA operator when $q_1 = 1$ and $q_2 = q_3 = \ldots = q_n = 0$:

$$AA(a_1, a_2, ..., a_n) = \frac{1}{n} \sum_{i=1}^{n} a_i \tag{4.4}$$

- The MM operator will reduce to the BM operator when q_1, $q_2 > 0$ and $q_3 = q_4 = \ldots = q_n = 0$:

$$BM^{(q_1, q_2)}(a_1, a_2, ..., a_n) = \left(\frac{1}{n(n-1)} \sum_{\substack{i,j=1 \\ j \neq i}}^{n} \left(a_i^{q_1} a_j^{q_2} \right) \right)^{\frac{1}{q_1 + q_2}} \tag{4.5}$$

- The MM operator will reduce to the MSM operator when $q_1 = q_2 = \ldots = q_k = 1$ and $q_{k+1} = q_{k+2} = \ldots = q_n = 0$:

$$MSM^{(k)}(a_1, a_2, ..., a_n) = \left(\left(\frac{k!(n-k)!}{n!} \right) \sum_{1 \leq i_1 < \ldots < i_k \leq n} \prod_{j=1}^{k} a_{i_j} \right)^{1/k} \tag{4.6}$$

- The MM operator will reduce to the GA operator when $q_1 = q_2 = \ldots = q_n = 1$:

$$GA(a_1, a_2, ..., a_n) = \left(\prod_{i=1}^{n} a_i \right)^{1/n} \tag{4.7}$$

4.1.3 Geometric MM (GMM) operator

The GMM operator is the dual form of the MM operator. This operator is also capable of capturing the interrelationships of multiple aggregated arguments and can provide a general form of the GA operator, geometric Bonferroni mean (GBM) operator, geometric Maclaurin symmetric mean (GMSM) operator and AA operator. Its formal definition is as follow:

Definition 4.7 (GMM operator). Let a_1, a_2, …, a_n be n positive numbers, $Q = (q_1, q_2, \ldots, q_n)$ be a collection of n real numbers such that $q_1, q_2, \ldots, q_n \geq 0$ but not at the same time $q_1 = q_2 = \ldots = q_n = 0$, $p(i)$ be

a permutation of (1, 2,…, n) and $\boldsymbol{P}_n$ be the set of all permutations of (1, 2,…, n). Then the aggregation function

$$GMM^{Q}(a_1, a_2, ..., a_n) = \left(1 \Big/ \sum_{i=1}^{n} q_i\right)\left(\prod_{p \in \boldsymbol{P}_n} \sum_{i=1}^{n} \left(q_i a_{p(i)}\right)\right)^{\frac{1}{n!}} \tag{4.8}$$

is called the GMM operator.

In the GMM operator, whether the interrelationships of a_1, a_2, …, a_n are captured also relies on the values of q_1, q_2, …, q_n with the same cases as these values are in the MM operator.

In addition, the GMM operator will reduce to the GA, GBM, GMSM and AA operators when q_1, q_2, …, q_n take some special values:

- The GMM operator will reduce to the GA operator in Equation (4.7) when $q_1 = 1$ and $q_2 = q_3 = \ldots = q_n = 0$.
- The GMM operator will reduce to the GBM operator when q_1, $q_2 > 0$ and $q_3 = q_4 = \ldots = q_n = 0$:

$$GBM^{(q_1, q_2)}(a_1, a_2, ..., a_n) = \frac{1}{q_1 + q_2} \prod_{\substack{i, j=1 \\ j \neq i}}^{n} \left(q_1 a_i + q_2 a_j\right)^{\frac{1}{n(n-1)}} \tag{4.9}$$

- The GMM operator will reduce to the GMSM operator when $q_1 = q_2 = \ldots = q_k = 1$ and $q_{k+1} = q_{k+2} = \ldots = q_n = 0$:

$$GMSM^{(k)}(a_1, a_2, ..., a_n) = \frac{1}{k}\left(\prod_{1 \leq i_1 < \ldots < i_k \leq n} \sum_{j=1}^{k} a_{i_j}\right)^{\frac{k!(n-k)!}{n!}} \tag{4.10}$$

- The GMM operator will reduce to the AA operator in Equation (4.4) when $q_1 = q_2 = \ldots = q_n = 1$.

4.1.4 Power average operator

The PA operator was introduced by Yager (2001). It can assign weights to the aggregated arguments dynamically via calculating the degrees of support between them. This enables the operator to reduce the effect of extreme argument values on the aggregation result. The formal definition of the PA operator is as follow:

Definition 4.8 (PA operator). Let a_1, a_2, …, a_n be n positive numbers, $s(a_i, a_j) = 1 - d(a_i, a_j)$ (where i, $j = 1, 2, \ldots, n$ and $j \neq i$ and $d(a_i, a_j)$ is the distance between a_i and a_j) be the degree of support for a_i from a_j such that $0 \leq s(a_i, a_j) \leq 1$, $s(a_i, a_j) = s(a_j, a_i)$ and $s(a_i, a_j) \geq s(a_{i'}, a_{j'})$ if and only if $|a_i, a_j| \leq |a_{i'}, a_{j'}|$ (where i', $j' = 1, 2, \ldots, n$ and $j' \neq i'$), and

$$t(a_i) = \sum_{\substack{j=1 \\ j \neq i}}^{n} s(a_i,\ a_j) \tag{4.11}$$

Then the aggregation function

$$PA(a_1, a_2, ..., a_n) = \frac{\sum_{i=1}^{n} \left(\left(1 + t(a_i)\right) a_i \right)}{\sum_{i=1}^{n} \left(1 + t(a_i)\right)} \tag{4.12}$$

is called the PA operator.

4.1.5 Archimedean t-norm and t-conorm

A t-norm is a binary operation on [0, 1] which satisfies the conditions of commutativity, associativity, monotonicity and boundary (Klement et al., 2000). The dual notion of a t-norm is its conorm. A t-norm and its conorm can be formally defined as follows:

Definition 4.9 (T-norm). A t-norm is a function $T : [0, 1]^2 \rightarrow [0, 1]$ such that for all $x, y, z \in [0, 1]$: $T(x, y) = T(y, x)$; $T(x, T(y, z)) = T(T(x, y), z)$; If $y \leq z$, then $T(x, y) \leq T(x, z)$; $T(x, 1) = x$.

Definition 4.10 (T-conorm). If T is a t-norm, then its conorm $T^C : [0, 1]^2 \rightarrow [0, 1]$ is $T^C(x, y) = 1 - T(1-x, 1-y)$.

A t-norm T is called Archimedean if every sequence x_n (where $n = 1, 2, \ldots$ and $x_1 < 1$ and $x_{n+1} = T(x_n, x_n)$) converges to 0. The conorm of an Archimedean t-norm is called an Archimedean t-conorm. For an Archimedean t-norm T and its conorm T^C:

- If a function $f(t)$ ($t \in R$) is monotonically decreasing and satisfies the conditions that $f(t) : (0, 1] \rightarrow R^+$; $f^{-1}(t) : R^+ \rightarrow (0, 1]$; $\lim_{t \rightarrow \infty} f^{-1}(t) = 0$; and $f^{-1}(0) = 1$, then $f(t)$ can be used to generate T: $T(x, y) = f^{-1}(f(x) + f(y))$ and is called as an additive generator (AG) of T.
- If a function $g(t)$ ($t \in R$) is monotonically increasing and satisfies the conditions that $g(t) : (0, 1] \rightarrow R^+$; $g^{-1}(t) : R^+ \rightarrow (0, 1]$; $\lim_{t \rightarrow \infty} g^{-1}(t) = 1$; and $g^{-1}(0) = 0$, then $g(t)$ can be used to generate T^C: $T^C(x, y) = g^{-1}(g(x) + g(y))$ and is called as an AG of T^C.

According to the definition of the conorm of a t-norm, it can be derived that $f(t) = g(1-t)$.

During the past few decades, study on Archimedean t-norm and t-conorm and their AGs has received extensive attention. Various families of Archimedean t-norms and t-conorms have been presented within the academia. Four well-known families are the Algebraic t-norm and t-conorm (ATT), Einstein t-norm and t-conorm (ETT), Hamacher t-norm and t-conorm (HTT) and Frank t-norm and t-conorm (FTT) (Xia et al., 2012; Liu and Wang, 2019). The AGs and expressions of these Archimedean t-norms and t-conorms are listed in Table 4.1 and Table 4.2, respectively.

Table 4.1 AGs and expressions of four Archimedean t-norms

Archimedean t-norm	AG of the t-norm	Inverse function	Expression of the t-norm
Algebraic	$f(t) = -\mathrm{In}t$	$f^{-1}(t) = \mathrm{e}^{-t}$	$T_{\mathrm{A}}(x, y) = xy$
Einstein	$f(t) = \mathrm{In}\dfrac{2-t}{t}$	$f^{-1}(t) = \dfrac{2}{\mathrm{e}^{t}+1}$	$T_{\mathrm{E}}(x, y) = \dfrac{xy}{1+(1-x)(1-y)}$
Hamacher	$f(t) = \mathrm{In}\dfrac{\delta+(1-\delta)t}{t}, (\delta > 0)$	$f^{-1}(t) = \dfrac{\delta}{\mathrm{e}^{t}+\delta-1}$	$T_{\mathrm{H}}(x, y) = \dfrac{xy}{\delta+(1-\delta)(x+y-xy)}$
Frank	$f(t) = -\mathrm{In}\dfrac{\varepsilon-1}{\varepsilon^{t}-1}, (\varepsilon > 1)$	$f^{-1}(t) = \log_{\varepsilon}\dfrac{\mathrm{e}^{t}+\varepsilon-1}{\mathrm{e}^{t}}$	$T_{\mathrm{F}}(x, y) = \log_{\varepsilon}\left(1+\dfrac{(\varepsilon^{x}-1)(\varepsilon^{y}-1)}{\varepsilon-1}\right)$

Table 4.2 AGs and expressions of four Archimedean t-conorms

Archimedean t-conorm	AG of the t-conorm	Inverse function	Expression of the t-conorm
Algebraic	$g(t) = -\mathrm{In}(1-t)$	$g^{-1}(t) = 1-\mathrm{e}^{-t}$	$T_{\mathrm{A}}^{\mathrm{C}}(x, y) = x+y-xy$
Einstein	$g(t) = \mathrm{In}\dfrac{1+t}{1-t}$	$g^{-1}(t) = \dfrac{\mathrm{e}^{t}-1}{\mathrm{e}^{t}+1}$	$T_{\mathrm{E}}^{\mathrm{C}}(x, y) = \dfrac{x+y}{1+xy}$
Hamacher	$g(t) = \mathrm{In}\dfrac{\delta+(1-\delta)(1-t)}{1-t}$	$g^{-1}(t) = \dfrac{\mathrm{e}^{t}-1}{\mathrm{e}^{t}+\delta-1}$	$T_{\mathrm{H}}^{\mathrm{C}}(x, y) = \dfrac{x+y-xy-(1-\delta)xy}{1-(1-\delta)xy}$
Frank	$g(t) = -\mathrm{In}\dfrac{\varepsilon-1}{\varepsilon^{1-t}-1}$	$g^{-1}(t) = 1-\log_{\varepsilon}\dfrac{\mathrm{e}^{t}+\varepsilon-1}{\mathrm{e}^{t}}$	$T_{\mathrm{F}}^{\mathrm{C}}(x, y) = 1-\log_{\varepsilon}\left[1+\dfrac{(\varepsilon^{1-x}-1)(\varepsilon^{1-y}-1)}{\varepsilon-1}\right]$

4.1.6 Operational laws of fuzzy numbers

Based on Archimedean t-norm and t-conorm, a set of general and flexible operational laws of fuzzy numbers can be developed. Formally, let $\alpha = \langle\mu\rangle$, $\alpha_1 = \langle\mu_1\rangle$ and $\alpha_2 = \langle\mu_2\rangle$ be three fuzzy numbers and λ be a positive number. Then the sum and product operations between α_1 and α_2 based on the Archimedean t-norm $T(x, y) = f^{-1}(f(x) + f(y))$ and its t-conorm $T^{\mathrm{C}}(x, y) = g^{-1}(g(x) + g(y))$ and the multiplication and power operations between λ and α based on $T(x, y)$ and $T^{\mathrm{C}}(x, y)$ can be respectively performed using the following laws:

$$\alpha_1 \oplus \alpha_2 = \langle T^{\mathrm{C}}(\mu_1, \mu_2)\rangle = \langle g^{-1}(g(\mu_1) + g(\mu_2))\rangle \tag{4.13}$$

$$\alpha_1 \otimes \alpha_2 = \langle T(\mu_1, \mu_2)\rangle = \langle f^{-1}(f(\mu_1) + f(\mu_2))\rangle \tag{4.14}$$

$$\lambda\alpha = \langle g^{-1}(\lambda g(\mu))\rangle \tag{4.15}$$

$$\alpha^{\lambda} = \langle f^{-1}(\lambda f(\mu))\rangle \tag{4.16}$$

The operational laws above can be regarded as the special cases of the operational laws of intuitionistic fuzzy numbers based on Archimedean t-norm and t-conorm developed by Xia et al. (2012) and the operational laws of generalised orthopair fuzzy numbers (commonly known as q-rung orthopair fuzzy numbers) based on Archimedean t-norm and t-conorm developed by Liu and Wang (2019). It is easy to prove that these operational laws have the following algebraic properties:

$$\alpha_1 \oplus \alpha_2 = \alpha_2 \oplus \alpha_1 \tag{4.17}$$

$$\alpha_1 \otimes \alpha_2 = \alpha_2 \otimes \alpha_1 \tag{4.18}$$

$$\lambda(\alpha_1 \oplus \alpha_2) = \lambda\alpha_1 \oplus \lambda\alpha_2 \tag{4.19}$$

$$(\alpha_1 \otimes \alpha_2)^{\lambda} = \alpha_1^{\lambda} \otimes \alpha_2^{\lambda} \tag{4.20}$$

$$\lambda_1\alpha \oplus \lambda_2\alpha = (\lambda_1 + \lambda_2)\alpha \tag{4.21}$$

$$\alpha^{\lambda 1} \otimes \alpha^{\lambda 2} = \alpha^{\lambda 1 + \lambda 2} \tag{4.22}$$

where λ_1 and λ_2 are two positive numbers.

4.2 Fuzzy Archimedean weighted power MM operator

In this section, an FAWPMM operator is presented on the basis of the MM and PA operators of fuzzy numbers, weights and the operational laws of fuzzy numbers based on Archimedean t-norm and t-conorm. The formal definition of the FAWPMM operator is provided. Its general expression is established, property is explored, special cases are discussed and specific expressions are established.

The FAWPMM operator is a weighted power MM operator of fuzzy numbers, in which the operations are performed using the operational laws of fuzzy numbers based on Archimedean t-norm and t-conorm. The formal definition of this operator is as follow:

Definition 4.11 (FAWPMM operator). Let $\alpha_1 = \langle\mu_1\rangle$, $\alpha_2 = \langle\mu_2\rangle$, …, $\alpha_n = \langle\mu_n\rangle$ be n fuzzy numbers, $Q = (q_1, q_2, \ldots, q_n)$ be a collection of n real numbers such that $q_1, q_2, \ldots, q_n \geq 0$ but not at the same time $q_1 = q_2 = \ldots = q_n = 0$, $p(i)$ be a permutation of $(1, 2, \ldots, n)$, $\boldsymbol{P}_n$ be the set of all permutations of $(1, 2, \ldots, n)$, $w_1, w_2, \ldots, w_n$ be respectively the weights of $\alpha_1, \alpha_2, \ldots, \alpha_n$ such that $0 \leq w_1, w_2, \ldots, w_n \leq 1$ and $w_1 + w_2 + \ldots + w_n = 1$, $s(\alpha_i, \alpha_j) = 1 - d(\alpha_i, \alpha_j)$ (where $i, j = 1, 2, \ldots, n$ and $j \neq i$ and $d(\alpha_i, \alpha_j)$ is the distance between α_i and α_j) be the degree of support for α_i from α_j such that $0 \leq s(\alpha_i, \alpha_j) \leq 1$, $s(\alpha_i, \alpha_j) = s(\alpha_j, \alpha_i)$ and $s(\alpha_i, \alpha_j) \geq s(\alpha_{i'}, \alpha_{j'})$ if and only if $|\alpha_i, \alpha_j| \leq |\alpha_{i'}, \alpha_{j'}|$ (where $i', j' = 1, 2, \ldots, n$ and $j' \neq i'$), and

$$t(\alpha_i) = \sum_{\substack{j=1 \\ j \neq i}}^{n} s(\alpha_i, \alpha_j) \tag{4.23}$$

Then the aggregation function

$$FAWPMM^{Q}(\alpha_1, \alpha_2, \ldots, \alpha_n) = \left(\frac{1}{n!} \bigoplus_{p \in \boldsymbol{P}_n} \bigotimes_{i=1}^{n} \left(\frac{n w_{p(i)} \left(1 + t(\alpha_{p(i)})\right)}{\sum_{j=1}^{n} w_j \left(1 + t(\alpha_j)\right)} \alpha_{p(i)} \right)^{q_i} \right)^{\frac{1}{\sum_{i=1}^{n} q_i}} \tag{4.24}$$

is called the FAWPMM operator.

In the FAWPMM operator, whether the interrelationships of $\alpha_1, \alpha_2, \ldots, \alpha_n$ are captured relies on the values of $q_1, q_2, \ldots, q_n$: If $q_1 > 0$ and $q_2 = q_3 = \ldots = q_n = 0$, then the interrelationships are not captured; If $q_1, q_2 > 0$ and $q_3 = q_4 = \ldots = q_n = 0$, then the interrelationships between two of $\alpha_1, \alpha_2, \ldots, \alpha_n$ are captured; If $q_1, q_2, \ldots, q_k > 0$ ($k = 3, 4, \ldots, n$) and $q_{k+1} = q_{k+2} = \ldots = q_n = 0$, then the interrelationships among k of $\alpha_1, \alpha_2, \ldots, \alpha_n$ are captured. In the following theorem, the general expression of the FAWPMM operator is established:

Theorem 4.1 (General expression). According to the expression in Equation (4.24) and the operational laws in Equation (4.13), Equation (4.14), Equation (4.15) and Equation (4.16), the FAWPMM operator can be further expressed as:

$$FAWPMM^{Q}(\alpha_1, \alpha_2, ..., \alpha_n) = \left\langle f^{-1}\left(\left(1\Big/\sum_{i=1}^{n} q_i\right) f\left(g^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} g\left(f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(\mu_{p(i)})\right)\right)\right)\right)\right)\right)\right)\right\rangle \tag{4.25}$$

where

$$\varphi_{p(i)} = \frac{w_{p(i)}\left(1+t(\alpha_{p(i)})\right)}{\sum_{j=1}^{n} w_j\left(1+t(\alpha_j)\right)} \tag{4.26}$$

and $FAWPMM^{Q}(\alpha_1, \alpha_2, \ldots, \alpha_n)$ is still a fuzzy number.

The proof of this theorem is given in Appendix A. The following theorem states the boundedness of the FAWPMM operator:

Theorem 4.2 (Boundedness). Let $\alpha_1 = \langle\mu_1\rangle$, $\alpha_2 = \langle\mu_2\rangle$, …, $\alpha_n = \langle\mu_n\rangle$ be n fuzzy numbers and $Q = (q_1, q_2, \ldots, q_n)$ be a collection of n real numbers such that $q_1, q_2, \ldots, q_n \geq 0$ but not at the same time $q_1 = q_2 = \ldots = q_n = 0$. If $\alpha^- = \langle\min(\mu_i)\rangle$ and $\alpha^+ = \langle\max(\mu_i)\rangle$ for all $i = 1, 2, \ldots, n$, then $\alpha^- \leq FAWPMM^{Q}(\alpha_1, \alpha_2, \ldots, \alpha_n) \leq \alpha^+$.

The proof of this theorem is given in Appendix B. It is worth noting that the FAPWMM operator does not have the properties of idempotency and commutativity because of the combination of weights of the aggregated fuzzy numbers. The FAWPMM operator will reduce to several other AOs when $q_1, q_2, \ldots, q_n$ take some special values:

- If $q_1 = 1$ and $q_2 = q_3 = \ldots = q_n = 0$, the FAWPMM operator will reduce to a fuzzy Archimedean weighted power arithmetic average (FAWPAA) operator:

$$FAWPAA(\alpha_1, \alpha_2, ..., \alpha_n) = g^{-1}\left(\frac{1}{n}\sum_{i=1}^{n}\left((n\varphi_i)g(\mu_i)\right)\right) \tag{4.27}$$

- If $q_1, q_2 > 0$ and $q_3 = q_4 = \ldots = q_n = 0$, the FAWPMM operator will reduce to a fuzzy Archi-

medean weighted power Bonferroni mean (FAWPBM) operator:

$$FAWPBM^{(q_1,q_2)}(\alpha_1,\alpha_2,...,\alpha_n) = \left\langle f^{-1}\left(\frac{1}{q_1+q_2} f\left(g^{-1}\left(\frac{1}{n(n-1)}\sum_{\substack{i,j=1\\ j\neq i}}^{n} g\left(f^{-1}\left(\begin{matrix} q_1 f\left(g^{-1}\left((n\varphi_i)g(\mu_i)\right)\right)+ \\ q_2 f\left(g^{-1}\left((n\varphi_j)g(\mu_j)\right)\right)\end{matrix}\right)\right)\right)\right)\right)\right\rangle \quad (4.28)$$

- If $q_1 = q_2 = \ldots = q_k = 1$ and $q_{k+1} = q_{k+2} = \ldots = q_n = 0$, the FAWPMM operator will reduce to a fuzzy Archimedean weighted power Maclaurin symmetric mean (FAWPMSM) operator:

$$FAWPMSM^{(k)}(\alpha_1,\alpha_2,...,\alpha_n) = \left\langle f^{-1}\left(\frac{1}{k} f\left(g^{-1}\left(\frac{k!(n-k)!}{n!}\sum_{1\le i_1<\ldots<i_k\le n} g\left(f^{-1}\left(\sum_{j=1}^{k}\left(f\left(g^{-1}\left((n\varphi_{i_j})g(\mu_{i_j})\right)\right)\right)\right)\right)\right)\right)\right)\right\rangle \quad (4.29)$$

- If $q_1 = q_2 = \ldots = q_n = 1$, the FAWPMM operator will reduce to a fuzzy Archimedean weighted power geometric average (FAWPGA) operator:

$$FAWPGA(\alpha_1,\alpha_2,...,\alpha_n) = f^{-1}\left(\frac{1}{n}\sum_{i=1}^{n}\left((n\varphi_i) f(\mu_i)\right)\right) \quad (4.30)$$

Equation (4.25) is a general form of the FAWPMM operator and cannot be directly used to aggregate fuzzy numbers. If the AGs of specific families of Archimedean t-norm and t-conorm are applied to the equation, then specific expressions of the FAWPMM operator, which can be directly used to aggregate fuzzy numbers, can be constructed. Four examples of specific expressions are as follows:

- If the AGs of ATT in Table 4.1 and Table 4.2 are applied to Equation (4.25), an FAWPMM operator based on ATT is established as follow:

$$FAWPMM^{Q}_{\mathrm{A}}(\alpha_1,\alpha_2,...,\alpha_n) = \left\langle \left(1-\prod_{p\in \boldsymbol{P}_n}\left(1-\prod_{i=1}^{n}\left(1-\left(1-\mu_{p(i)}\right)^{n\varphi_{p(i)}}\right)^{q_i}\right)^{\frac{1}{n!}}\right)^{\frac{1}{\sum_{i=1}^{n} q_i}}\right\rangle \quad (4.31)$$

- If the AGs of ETT in Table 4.1 and Table 4.2 are applied to Equation (4.25), an FAWPMM operator based on ETT is established as follow:

$$FAWPMM^{Q}_{\mathrm{E}}(\alpha_1,\alpha_2,...,\alpha_n) = \left\langle \frac{\left(2\left(\mu'-1\right)^{1/\sum_{i=1}^{n} q_i}\right)}{\left(\left(\mu'+3\right)^{1/\sum_{i=1}^{n} q_i}+\left(\mu'-1\right)^{1/\sum_{i=1}^{n} q_i}\right)}\right\rangle \quad (4.32)$$

where

$$\mu' = \prod_{p\in \boldsymbol{P}_n}\left(\frac{\prod_{i=1}^{n}\left(\frac{\left(1+\mu_{p(i)}\right)^{n\varphi_{p(i)}}+3\left(1-\mu_{p(i)}\right)^{n\varphi_{p(i)}}}{\left(1+\mu_{p(i)}\right)^{n\varphi_{p(i)}}-\left(1-\mu_{p(i)}\right)^{n\varphi_{p(i)}}}\right)^{q_i}+3}{\prod_{i=1}^{n}\left(\frac{\left(1+\mu_{p(i)}\right)^{n\varphi_{p(i)}}+3\left(1-\mu_{p(i)}\right)^{n\varphi_{p(i)}}}{\left(1+\mu_{p(i)}\right)^{n\varphi_{p(i)}}-\left(1-\mu_{p(i)}\right)^{n\varphi_{p(i)}}}\right)^{q_i}-1}\right)^{\frac{1}{n!}} \tag{4.33}$$

- If the AGs of HTT in Table 4.1 and Table 4.2 are applied to Equation (4.25), an FAWPMM operator based on HTT is established as follow:

$$FAWPMM_{\mathrm{H}}^{Q}(\alpha_1,\alpha_2,...,\alpha_n) = \left\langle \frac{\left(\delta\left(\mu'-1\right)^{1/\sum_{i=1}^{n}q_i}\right)}{\left(\left(\mu'+\delta^2-1\right)^{1/\sum_{i=1}^{n}q_i}+(\delta-1)\left(\mu'-1\right)^{1/\sum_{i=1}^{n}q_i}\right)}\right\rangle \tag{4.34}$$

where

$$\mu' = \prod_{p\in \boldsymbol{P}_n}\left(\frac{\prod_{i=1}^{n}\left(\frac{\left(\delta+(1-\delta)(1-\mu_{p(i)})\right)^{n\varphi_{p(i)}}+(\delta^2-1)\left(1-\mu_{p(i)}\right)^{n\varphi_{p(i)}}}{\left(\delta+(1-\delta)(1-\mu_{p(i)})\right)^{n\varphi_{p(i)}}-\left(1-\mu_{p(i)}\right)^{n\varphi_{p(i)}}}\right)^{q_i}+\delta^2-1}{\prod_{i=1}^{n}\left(\frac{\left(\delta+(1-\delta)(1-\mu_{p(i)})\right)^{n\varphi_{p(i)}}+(\delta^2-1)\left(1-\mu_{p(i)}\right)^{n\varphi_{p(i)}}}{\left(\delta+(1-\delta)(1-\mu_{p(i)})\right)^{n\varphi_{p(i)}}-\left(1-\mu_{p(i)}\right)^{n\varphi_{p(i)}}}\right)^{q_i}-1}\right)^{\frac{1}{n!}} \tag{4.35}$$

- If the AGs of FTT in Table 4.1 and Table 4.2 are applied to Equation (4.25), an FAWPMM operator based on FTT is established as follow:

$$FAWPMM_{\mathrm{F}}^{Q}(\alpha_1,\alpha_2,...,\alpha_n) = \left\langle \log_{\varepsilon}\left(1+\frac{\left(\varepsilon^{\mu'''}-1\right)^{1/\sum_{i=1}^{n}q_i}}{\left(\varepsilon-1\right)^{1/\sum_{i=1}^{n}q_i-1}}\right)\right\rangle \tag{4.36}$$

where

$$\mu''' = 1-\log_{\varepsilon}\left(1+(\varepsilon-1)\Big/\prod_{p\in \boldsymbol{P}_n}\left(\frac{\varepsilon-1}{\varepsilon^{1-\mu''}-1}\right)^{\frac{1}{n!}}\right) \tag{4.37}$$

$$\mu'' = \log_{\varepsilon}\left(1+(\varepsilon-1)\Big/\prod_{i=1}^{n}\left(\frac{\varepsilon-1}{\varepsilon^{\mu'}-1}\right)^{q_i}\right) \tag{4.38}$$

$$\mu' = 1-\log_{\varepsilon}\left(1+\frac{\left(\varepsilon^{1-\mu_{p(i)}}-1\right)^{n\varphi_{p(i)}}}{\left(\varepsilon-1\right)^{n\varphi_{p(i)}-1}}\right) \tag{4.39}$$

4.3 Fuzzy Archimedean weighted power GMM operator

In this section, an FAWPGMM operator is presented on the basis of the GMM and PA operators of fuzzy numbers, weights and the operational laws of fuzzy numbers based on Archimedean t-norm and t-conorm. The formal definition of the FAWPGMM operator is provided. Its general expression is established, properties are explored, special cases are discussed and specific expressions are established.

The FAWPGMM operator is a weighted power GMM operator of fuzzy numbers, in which the operations are performed using the operational laws of fuzzy numbers based on Archimedean t-norm and t-conorm. The formal definition of this operator is as follow:

Definition 4.12 (FAWPGMM operator). Based on Definition 4.11, the aggregation function

$$FAWPGMM^{Q}(\alpha_1,\alpha_2,...,\alpha_n)=\left(1\Big/\sum_{i=1}^{n}q_i\right)\left(\bigotimes_{p\in \boldsymbol{P}_n}\bigoplus_{i=1}^{n}\left(q_i\alpha_{p(i)}^{n\varphi_{p(i)}}\right)\right)^{\frac{1}{n!}} \tag{4.40}$$

is called the FAWPGMM operator.

In the FAWPGMM operator, whether the interrelationships of α_1, α_2, …, α_n are captured also relies on the values of q_1, q_2, …, q_n: If $q_1 > 0$ and $q_2 = q_3 = \ldots = q_n = 0$, then the interrelationships are not captured; If q_1, $q_2 > 0$ and $q_3 = q_4 = \ldots = q_n = 0$, then the interrelationships between two of α_1, α_2, …, α_n are captured; If q_1, q_2, …, $q_k > 0$ (k = 3, 4, …, n) and $q_{k+1} = q_{k+2} = \ldots = q_n = 0$, then the interrelationships among k of α_1, α_2, …, α_n are captured. In the following theorem, the general expression of the FAWPGMM operator is established:

Theorem 4.3 (General expression). According to the expression in Equation (4.40) and the operational laws in Equation (4.13), Equation (4.14), Equation (4.15) and Equation (4.16), the FAWPGMM operator can be further expressed as:

$$FAWPGMM^{Q}(\alpha_1,\alpha_2,...,\alpha_n)=$$
$$\left\langle g^{-1}\left(\left(1\Big/\sum_{i=1}^{n}q_i\right)g\left(f^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n}f\left(g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)})f(\mu_{p(i)})\right)\right)\right)\right)\right)\right)\right)\right\rangle \tag{4.41}$$

where $\varphi_{p(i)}$ is given in Equation (4.26) and $FAWPGMM^{Q}(\alpha_1, \alpha_2, \ldots, \alpha_n)$ is still a fuzzy number.

The proof of this theorem is given in Appendix C. The following theorem states the boundedness of the FAWPGMM operator:

Theorem 4.4 (Boundedness). Let $\alpha_1=\langle\mu_1\rangle$, $\alpha_2=\langle\mu_2\rangle$, …, $\alpha_n=\langle\mu_n\rangle$ be n fuzzy numbers and $Q = (q_1, q_2, \ldots, q_n)$ be a collection of n real numbers such that $q_1, q_2, \ldots, q_n \geq 0$ but not at the same time $q_1 = q_2 = \ldots = q_n = 0$. If $\alpha^{-} = \langle\min(\mu_i)\rangle$ and $\alpha^{+} = \langle\max(\mu_i)\rangle$ for all i = 1, 2, …, n, then $\alpha^{-} \leq FAWPGMM^{Q}(\alpha_1, \alpha_2, \ldots, \alpha_n) \leq \alpha^{+}$.

The proof of this theorem is given in Appendix D. It is worth noting that the FAWPGMM operator does not have the properties of idempotency and commutativity due to the combination of weights of the aggregated fuzzy numbers. The FAWPGMM operator will reduce to several other AOs when $q_1, q_2, \ldots, q_n$ take some special values:

- If $q_1 = 1$ and $q_2 = q_3 = \ldots = q_n = 0$, then the FAWPGMM operator will reduce to the FAWP-GA operator in Equation (4.30).
- If $q_1, q_2 > 0$ and $q_3 = q_4 = \ldots = q_n = 0$, then the FAWPGMM operator will reduce to a fuzzy Archimedean weighted power geometric Bonferroni mean (FAWPGBM) operator:

$$FAWPGBM^{(q_1,q_2)}(\alpha_1,\alpha_2,...,\alpha_n) = \left\langle g^{-1}\left(\frac{1}{q_1+q_2}g\left(f^{-1}\left(\frac{1}{n(n-1)}\sum_{\substack{i,j=1\\ j\neq i}}^{n}\left(f\left(g^{-1}\begin{pmatrix} q_1 g\left(f^{-1}\left((n\varphi_i)f(\mu_i)\right)\right)+ \\ q_2 g\left(f^{-1}\left((n\varphi_j)f(\mu_j)\right)\right)\end{pmatrix}\right)\right)\right)\right)\right)\right\rangle \quad (4.42)$$

- If $q_1 = q_2 = \ldots = q_k = 1$ and $q_{k+1} = q_{k+2} = \ldots = q_n = 0$, then the FAWPGMM operator will reduce to a fuzzy Archimedean weighted power geometric Maclaurin symmetric mean (FAW-PGMSM) operator:

$$FAWPGMSM^{(k)}(\alpha_1,\alpha_2,...,\alpha_n) = \left\langle g^{-1}\left(\frac{1}{k}g\left(f^{-1}\left(\frac{k!(n-k)!}{n!}\sum_{1\le i_1<\ldots<i_k\le n}f\left(g^{-1}\left(\sum_{j=1}^{k}\left(g\left(f^{-1}\left((n\varphi_{i_j})f(\mu_{i_j})\right)\right)\right)\right)\right)\right)\right)\right)\right\rangle \quad (4.43)$$

- If $q_1 = q_2 = \ldots = q_n = 1$, then the FAWPGMM operator will reduce to the FAWPAA operator in Equation (4.27).

Equation (4.41) is a general form of the FAWPGMM operator and cannot be directly used to aggregate fuzzy numbers. If the AGs of specific families of Archimedean t-norm and t-conorm are applied to the equation, then specific expressions of the FAWPGMM operator, which can be directly used to aggregate fuzzy numbers, can be constructed. Four examples of specific expressions are as follows:

- If the AGs of ATT in Table 4.1 and Table 4.2 are applied to Equation (4.41), an FAWPGMM operator based on ATT is established as follow:

$$FAWPGMM_{\mathrm{A}}^{Q}(\alpha_1,\alpha_2,...,\alpha_n) = \left\langle 1-\left(1-\prod_{p\in \boldsymbol{P_n}}\left(1-\prod_{i=1}^{n}\left(1-\mu_{p(i)}^{n\varphi_{p(i)}}\right)^{q_i}\right)^{\frac{1}{n!}}\right)^{\frac{1}{\sum_{i=1}^{n}q_i}}\right\rangle \quad (4.44)$$

- If the AGs of ETT in Table 4.1 and Table 4.2 are applied to Equation (4.41), an FAWPGMM

operator based on ETT is established as follow:

$$FAWPGMM_{\mathrm{E}}^{Q}(\alpha_1,\alpha_2,...,\alpha_n)=\left\langle\frac{\left(\left(\mu'+3\right)^{1/\sum_{i=1}^{n}q_i}-\left(\mu'-1\right)^{1/\sum_{i=1}^{n}q_i}\right)}{\left(\left(\mu'+3\right)^{1/\sum_{i=1}^{n}q_i}+\left(\mu'-1\right)^{1/\sum_{i=1}^{n}q_i}\right)}\right\rangle \tag{4.45}$$

where

$$\mu'=\prod_{p\in\boldsymbol{P}_n}\left(\frac{\prod_{i=1}^{n}\left(\frac{\left(2-\mu_{p(i)}\right)^{n\varphi_{p(i)}}+3\mu_{p(i)}^{n\varphi_{p(i)}}}{\left(2-\mu_{p(i)}\right)^{n\varphi_{p(i)}}-\mu_{p(i)}^{n\varphi_{p(i)}}}\right)^{q_i}+3}{\prod_{i=1}^{n}\left(\frac{\left(2-\mu_{p(i)}\right)^{n\varphi_{p(i)}}+3\mu_{p(i)}^{n\varphi_{p(i)}}}{\left(2-\mu_{p(i)}\right)^{n\varphi_{p(i)}}-\mu_{p(i)}^{n\varphi_{p(i)}}}\right)^{q_i}-1}\right)^{\frac{1}{n!}} \tag{4.46}$$

- If the AGs of HTT in Table 4.1 and Table 4.2 are applied to Equation (4.41), an FAWPGMM operator based on HTT is established as follow:

$$FAWPGMM_{\mathrm{H}}^{Q}(\alpha_1,\alpha_2,...,\alpha_n)=\left\langle\frac{\left(\mu'+\delta^2-1\right)^{1/\sum_{i=1}^{n}q_i}-\left(\mu'-1\right)^{1/\sum_{i=1}^{n}q_i}}{\left(\mu'+\delta^2-1\right)^{1/\sum_{i=1}^{n}q_i}+(\delta-1)\left(\mu'-1\right)^{1/\sum_{i=1}^{n}q_i}}\right\rangle \tag{4.47}$$

where

$$\mu'=\prod_{p\in\boldsymbol{P}_n}\left(\frac{\prod_{i=1}^{n}\left(\frac{\left(\delta+(1-\delta)\mu_{p(i)}\right)^{n\varphi_{p(i)}}+(\delta^2-1)\mu_{p(i)}^{n\varphi_{p(i)}}}{\left(\delta+(1-\delta)\mu_{p(i)}\right)^{n\varphi_{p(i)}}-\mu_{p(i)}^{n\varphi_{p(i)}}}\right)^{q_i}+\delta^2-1}{\prod_{i=1}^{n}\left(\frac{\left(\delta+(1-\delta)\mu_{p(i)}\right)^{n\varphi_{p(i)}}+(\delta^2-1)\mu_{p(i)}^{n\varphi_{p(i)}}}{\left(\delta+(1-\delta)\mu_{p(i)}\right)^{n\varphi_{p(i)}}-\mu_{p(i)}^{n\varphi_{p(i)}}}\right)^{q_i}-1}\right)^{\frac{1}{n!}} \tag{4.48}$$

- If the AGs of FTT in Table 4.1 and Table 4.2 are applied to Equation (4.41), an FAWPGMM operator based on FTT is established as follow:

$$FAWPGMM_{\mathrm{F}}^{Q}(\alpha_1,\alpha_2,...,\alpha_n)=\left\langle 1-\log_{\varepsilon}\left(1+\frac{\left(\varepsilon^{1-\mu'''}-1\right)^{1/\sum_{i=1}^{n}q_i}}{\left(\varepsilon-1\right)^{1/\sum_{i=1}^{n}q_i-1}}\right)\right\rangle \tag{4.49}$$

where

$$\mu'''=\log_{\varepsilon}\left(1+(\varepsilon-1)\Big/\prod_{p\in\boldsymbol{P}_n}\left(\frac{\varepsilon-1}{\varepsilon^{\mu''}-1}\right)^{\frac{1}{n!}}\right) \tag{4.50}$$

$$\mu'' = 1 - \log_{\varepsilon}\left(1 + (\varepsilon - 1) \Big/ \prod_{i=1}^{n}\left(\frac{\varepsilon - 1}{\varepsilon^{1-\mu'} - 1}\right)^{q_i}\right) \tag{4.51}$$

$$\mu' = \log_{\varepsilon}\left(1 + \frac{\left(\varepsilon^{\mu_{p(i)}} - 1\right)^{n\varphi_{p(i)}}}{\left(\varepsilon - 1\right)^{n\varphi_{p(i)} - 1}}\right) \tag{4.52}$$

4.4 Summary

This chapter has constructed an FAWPMM operator and an FAWPGMM operator. The formal definitions of these operators have been presented. Their general expressions have been established via the operational laws of fuzzy numbers based on Archimedean t-norm and t-conorm. The properties of the two operators have been explored. Their special cases have been discussed. The specific expressions of the two operators have been established through applying the AGs of ATT, ETT, HTT and FTT. The constructed operators will be applied in development of a novel MADM method for automatic selection of the OBO of an LPBF part in Chapter 5.

5 The proposed automatic selection method

In this chapter, a novel MADM method for automatic selection of the OBO of an LPBF part is presented. The schematic diagram of the presented method is shown in Figure 5.1. The method mainly includes estimation of attribute values, normalisation of attribute values and selection of the OBO. In the estimation of attribute values, the STL model and ABOs of an LPBF part, the material and process parameters used to build the part and the attributes of ABOs are used as the input of certain estimation models of attribute values. The values of the attributes of ABOs are calculated by the estimation models. In the normalisation of attribute values, a ratio model is introduced to fuzzify the estimated attribute values and a fuzzy decision matrix is established and then normalised. In the selection of the OBO, the relationships and weights of the attributes of ABOs are determined. Based on them, the constructed fuzzy AOs in Chapter 4 are used to aggregate the fuzzy information in the normalised fuzzy decision matrix. A ranking of the generated ABOs is obtained via comparing the aggregation result and then the OBO of the part can be selected according to the ranking.

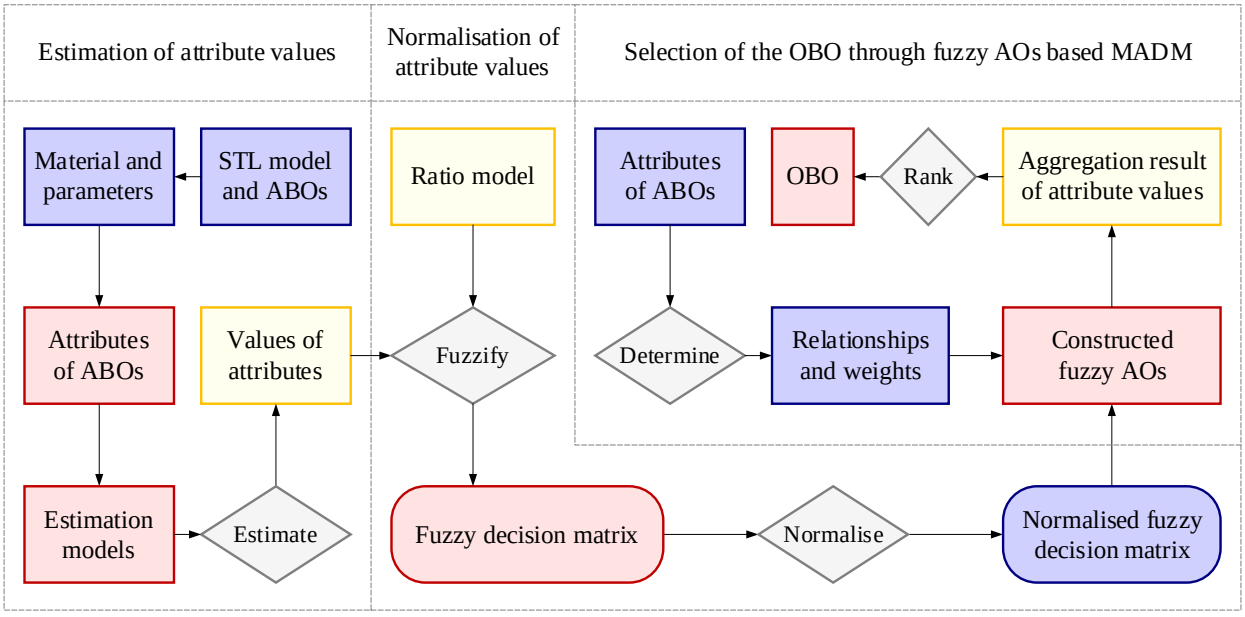


Figure 5.1 Schematic diagram of the automatic selection method

The chapter is organised as follows. Section 5.1 describes the specific process of the estimation of attribute values. The details of the normalisation of attribute values and the selection of the OBO are explained in Section 5.2 and Section 5.3, respectively. Section 5.4 ends the chapter with a summary.

5.1 Estimation of attribute values

According to the studies of Edwards and Ramulu (2014), Wauthle et al. (2015), Brika et al. (2017), Cheng and To (2019) and Griffiths et al. (2019), the attributes of an LPBF part affected by the build orientation of the part are support volume, volumetric error, surface roughness, build time, build cost, strength, elongation, hardness, residual stress, flexural modulus and fatigue performance. Among them, the support volume, volumetric error, surface roughness, build time and build cost in a given build orientation can be estimated via analysing the geometry of the STL model of the part:

- Support volume. When the angle between the normal vector of a planar triangular facet of the STL model of an LPBF part and the build orientation is greater than a threshold (commonly 135°), this facet is considered as an overhang and additional supports are required to sustain the overhang to avoid collapse or build failure during the fabrication of the part. Apart from overhangs, supports are also needed for the top areas of a hollow 3D model. In process planning for LPBF AM, the total volume of the supports required for a part is an important factor, as it will affect the build time and build cost of the part. To estimate the total support volume of an LPBF part, Autodesk Meshmixer, a free software tool for working with triangle meshes, is applied in the proposed method. Firstly, the STL model of an LPBF part is imported into the software tool. Then the model is rotated around the x axis by an angle of θ_x and rotated around the y axis by an angle of θ_y, so that the build orientation O given by a unitised vector (x, y, z) is vertically upward. After the rotations, supports can be generated and their volume (mm^3) can be estimated by the software tool. In this process, the angles (deg) θ_x and θ_y can be obtained via the following equation (Zhang et al., 2019):

$$\begin{cases} \theta_x = \arctan\left(y/z\right) & \text{if } y \neq 0 \text{ or } z \neq 0 \\ \theta_y = -\arctan\left(x/\sqrt{y^2+z^2}\right) \end{cases} \tag{5.1}$$

  It is worth noting that $\theta_x = 0$ deg when $y = z = 0$. In theory, the greater the total support volume, the longer the build time, and the higher the build cost. The objective is to minimise the total support volume if it is one of the attributes considered in selection of the OBO.

- Volumetric error. There are two types of surface errors in LPBF AM. The first type appears in the tessellation of CAD models. It is caused by approximating a surface with a set of planar triangles and can be reduced via controlling the chordal error. The second type is the so-called volumetric error, which is caused by fabricating each facet with staircases in a layer by layer manner and may have significant influence on the shape accuracy of the built part. Vol-

umetric error cannot be eliminated, but its effect can be reduced through specifying appropriate build orientation and layer thickness. To date, a number of models for estimating the total volumetric error of an AM part based on the STL model, build orientation and layer thickness of the part have been presented (Masood et al., 2000; 2003; Li and Zhang, 2013; Zhang and Li, 2013; Luo and Wang, 2016; Matos et al., 2020). The model of Luo and Wang (2016) is used to estimate the total volumetric error of an LPBF part in a given build orientation O. Firstly, the volumetric error of each facet in the STL model of an LPBF part in O is estimated via geometric analysis. The total volumetric error of the part in O is then obtained by calculating the sum of the volumetric error of all facets. Suppose the STL model of an LPBF part includes n facets $F_0 = (v_{0,1}, v_{0,2}, v_{0,3})$, $F_1 = (v_{1,1}, v_{1,2}, v_{1,3})$, …, $F_{n-1} = (v_{n-1,1}, v_{n-1,2}, v_{n-1,3})$. Then the total volumetric error (mm^3) of this part in O, as illustrated in Figure 5.2, can be estimated using the following equation:

$$E^{O}_{\text{Volumetric}} = \sum_{i=0}^{n-1} E_i = \sum_{i=0}^{n-1} \left(\frac{t_{\text{Layer}}}{2} \left| \cos\theta_i \right| A_i \right) \tag{5.2}$$

where E_i is the volumetric error (mm^3) of F_i, t_{Layer} is the layer thickness (mm) under the uniform planar slicing in O, θ_i is the angle (deg) between the normal vector of F_i and O, and A_i is the area (mm^2) of F_i that can be calculated using the following equation:

$$A_i = \sqrt{p_i \left(p_i - d(v_{i,1}, v_{i,2}) \right) \left(p_i - d(v_{i,1}, v_{i,3}) \right) \left(p_i - d(v_{i,2}, v_{i,3}) \right)} \tag{5.3}$$

where $d(v_{i,1}, v_{i,2})$, $d(v_{i,1}, v_{i,3})$ and $d(v_{i,2}, v_{i,3})$ are respectively the lengths (mm) of the three edges of F_i that can be computed using Equation (3.2), and

$$p_i = \frac{1}{2} \left(d(v_{i,1}, v_{i,2}) + d(v_{i,1}, v_{i,3}) + d(v_{i,2}, v_{i,3}) \right) \tag{5.4}$$

Generally, the greater the total volumetric error of an LPBF part, the greater the impact on the shape accuracy of the part. Thus, the objective is to minimise the total volumetric error if it is one of the attributes considered in selection of the OBO.

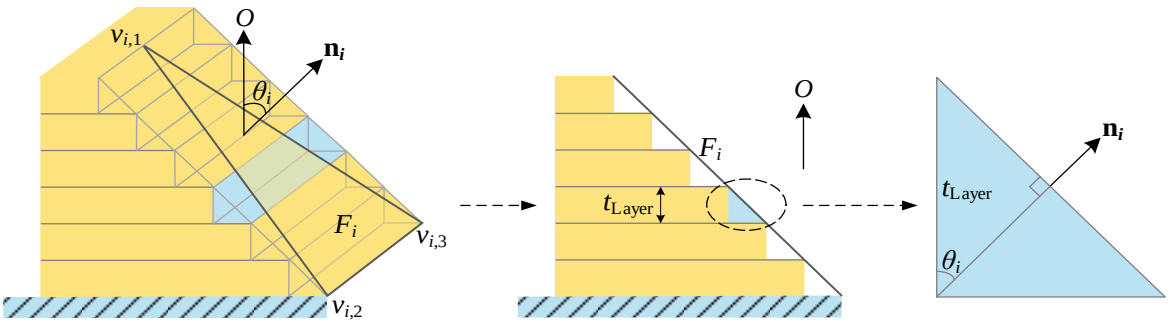


Figure 5.2 An illustration of the volumetric error of a facet

- Surface roughness. Surface roughness reflects the unevenness of a real surface. It is generally quantified by the deviations in the normal vector direction of the real surface from its ideal form. Small deviations indicate that the real surface is smooth. There are a number of different parameters available for describing surface roughness (ISO 4287, 1997), where *Ra* is the most common by far. In the field of AM, *Ra* is also the most commonly used parameter for measuring the surface roughness of an AM part (Strano et al., 2013; Calignano, 2018). Without loss of generality, this parameter will also be used to quantify the surface roughness of an LPBF part in this chapter and the next chapter. That is, all surface roughness values appearing in these two chapters are *Ra* values. In the literature, many estimation models of the average surface roughness of an AM part are established based on the staircase effect caused by the layer upon layer building manner of AM processes. However, according to the research findings of Strano et al. (2013) and Calignano (2018), the estimation models based on staircase effect cannot accurately predict the average surface roughness of an LPBF part, since it is mainly affected by build orientation and process parameters. In the MOO method of Brika et al. (2017), a model for estimating the average surface roughness of an LPBF part was established on the basis of a study of the average surface roughness of LPBF Ti6Al4V samples in different build orientations and with a constant layer thickness of 0.03 mm. This model is used to predict the average surface roughness of an LPBF part in a given build orientation O in the proposed method. Firstly, the surface roughness of each facet in the STL model of an LPBF part in O is estimated by a linear regression function. Then the roughness per unit area is calculated and used as the average surface roughness of the part in O. Suppose the STL model of an LPBF part includes n facets $F_0 = (v_{0,1}, v_{0,2}, v_{0,3})$, $F_1 = (v_{1,1}, v_{1,2}, v_{1,3})$, …, $F_{n-1} = (v_{n-1,1}, v_{n-1,2}, v_{n-1,3})$. The average surface roughness (mm) of this part in O can be estimated using the following equation:

$$Ra^O = \sum_{i=0}^{n-1}\left(Ra_i A_i\right) \Big/ \sum_{i=0}^{n-1} A_i = \sum_{i=0}^{n-1}\left((0.0094148 + 0.0000389\left|\theta_i - 90\right|)A_i\right) \Big/ \sum_{i=0}^{n-1} A_i \tag{5.5}$$

where Ra_i is the surface roughness (mm) of F_i in O, A_i is the area (mm^2) of F_i that can be calculated using Equation (5.3), and θ_i is value of the angle (deg) between the normal vector of F_i and O. For a built LPBF part, the greater the average surface roughness, the worse the surface quality. Therefore, the objective is to minimise the average surface roughness if it is one of the attributes considered in selection of the OBO.

- Build time. Build time and build cost are two important indicators for nearly all AM processes. Many of the existing OBO selection methods consider these factors and establish their es-

timation models with respect to build orientation and process parameters under specific AM processes. To estimate the build time of an LPBF part in a given build orientation *O*, an estimation model for the LPBF process presented by Brika et al. (2017) is introduced. According to the model, the total build time (s) of an LPBF part in *O* can be estimated using the following equation:

$$T_{\text{Build}}^{O} = \left(h_{\text{Model}}^{O} / t_{\text{Layer}}\right) T_{\text{Recoating}} + \left(V_{\text{Model}} + V_{\text{Support}}^{O}\right) / R_{\text{Build}} \tag{5.6}$$

where R_{Build} is the build rate (mm^3/s), i.e. the volume deposited per unit time, and other symbols are explained in Table 5.1. The build rate can be calculated using the following equation (Tang et al., 2017):

$$R_{\text{Build}} = t_{\text{Layer}} s_{\text{Hatch}} v_{\text{Scanning}} \tag{5.7}$$

where all symbols are also defined in Table 5.1. Obviously, the objective is to minimise the total build time if it is one of the attributes considered in selection of the OBO.

Table 5.1 An explanation of the symbols in the estimation model of build time

Symbol	Meaning	Unit	Value
h_{Model}^{O}	STL model height in *O*	mm	Calculate according to the geometry of the STL model and *O*
t_{Layer}	Layer thickness	mm	Specify according to the technical data of the used LPBF machine
$T_{\text{Recoating}}$	Recoating time	s	Specify according to the technical data of the used LPBF machine
V_{Model}	STL model volume	mm^3	Calculate according to the geometry of the STL model
V_{Support}^{O}	Support volume in *O*	mm^3	Estimate by the analysis module of Autodesk Meshmixer
s_{Hatch}	Hatch spacing	mm	Specify according to the technical data of the used LPBF machine
v_{Scanning}	Laser scanning velocity	mm/s	Specify according to the technical data of the used LPBF machine

- Build cost. To estimate the build cost of an LPBF part in a given build orientation, a generic estimation model presented by Baumers et al. (2013; 2016) is used. According to the model, build cost consists of direct cost and indirect cost, where the direct cost includes material cost and energy cost. That is, the total build cost (£) of an LPBF part in a given build orientation *O* is the sum of the material cost, energy cost and indirect cost of the part in *O*. This can be described by the following equation:

$$C_{\text{Build}}^{O} = C_{\text{Material}}^{O} + C_{\text{Energy}}^{O} + C_{\text{Indirect}}^{O} \tag{5.8}$$

  According to an estimation model of build cost for the LPBF process proposed by Brika et al. (2017), the material cost (£), energy cost (£) and indirect cost (£) in *O* can be calculated using the following equations:

$$C^{O}_{\text{Material}} = \left(V_{\text{Model}} + V^{O}_{\text{Support}}\right)(1 + r_{\text{Waste}})\rho_{\text{Material}} P_{\text{Material}} p_{\text{Material}} \tag{5.9}$$

$$C^{O}_{\text{Energy}} = \left(V_{\text{Model}} + V^{O}_{\text{Support}}\right)\rho_{\text{Material}} P_{\text{Material}} R_{\text{kWh/kg}} p_{\text{Energy}} \tag{5.10}$$

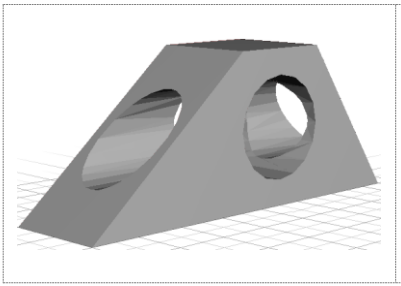
$$C^{O}_{\text{Indirect}} = R_{\text{£/h}} T^{O}_{\text{Build}} \tag{5.11}$$

where all symbols are explained in Table 5.2. Obviously, the objective is to minimise the total build cost if it is one of the attributes considered in selection of the OBO.

Table 5.2 An explanation of the symbols in the estimation model of build cost

Symbol	Meaning	Unit	Value
V_{Model}	STL model volume	m^3	Calculate according to the geometry of the STL model
V^{O}_{Support}	Support volume in *O*	m^3	Estimate by the analysis module of Autodesk Meshmixer
r_{Waste}	Material waste ratio	——	0.1, found from the data used in (Ruffo et al., 2006)
ρ_{Material}	Density of material	kg/m^3	Find from the technical data of the used material
P_{Material}	Porosity of material	——	Find from the technical data of the used material
p_{Material}	Price of material	£/kg	Find from the material supplier's website
$R_{\text{kWh/kg}}$	Energy consumption rate	kWh/kg	162.13, found from the data used in (Brika et al., 2017)
p_{Energy}	Price of energy	£/kWh	Find from the energy supplier's website
$R_{\text{£/h}}$	Indirect cost rate	£/h	44.08, converted from 53.35 $/h in (Brika et al., 2017)
T^{O}_{Build}	Build time in *O*	h	Calculate using the estimation model in Equation (5.6)

To illustrate the estimation of the support volume, volumetric error, surface roughness, build time and build cost under a given build orientation, a classic part for demonstrating a build orientation determination method is introduced. This part was developed in the method of Cheng et al. (1995) and has been applied to illustrate the methods of Pham et al. (1999), Pandey et al. (2004), Byun and Lee (2006), Canellidis et al. (2009), Zhang et al. (2016), Ransikarbum and Kim (2017) and Qie et al. (2018). The STL model of the part is shown in Figure 5.3. Suppose the part will be built using Ti6Al4V and the LPBF machine EOSINT M270. The ABOs of the part are generated by the proposed automatic generation method in Chapter 3 and are listed in Table 3.1 and shown in Figure 3.10. The values of layer thickness, recoating time, hatch spacing, laser scanning velocity, material density, material porosity and material price are cited from (Brika et al., 2017) and listed in Table 5.3. The value of energy price is obtained from the energy supplier's website and also provided in Table 5.3. Based on these conditions, the total support volume, total volumetric error, average surface roughness, total build time and total build cost of the LPBF Ti6Al4V part under each ABO can be estimated using the estimation models above. The estimation results are listed in Table 5.4.



Vertices:	134
Facets:	276
Surface:	Regular
Length:	75.00 mm
Width:	20.00 mm
Height:	30.00 mm
Area:	7,616.44 mm^2
Volume:	16,496.50 mm^3

Figure 5.3 The STL model of an LPBF part

Table 5.3 The values of some variables for estimating attributes values

Variable	t_{Layer}	$T_{Recoating}$	s_{Hatch}	$v_{Scanning}$	$\rho_{Material}$	$P_{Material}$	$p_{Material}$	p_{Energy}
Value	0.03	20	0.07	1,250	4,430	99.5%	247.86	0.16
Unit	mm	s	mm	mm/s	kg/m^3	——	£/kg	£/kWh

Table 5.4 The estimated attribute values in the generated ABOs

The generated ABO in unit vector	**Support volume** (unit: mm^3)	**Volumetric error** (unit: mm^3)	**Surface roughness** (unit: μm)	**Build time** (unit: h)	**Build cost** (unit: £)
O_1 = (00.0000, −1.0000, 00.0000)	5,712.6000	69.4984	11.1420	7.9057	377.7143
O_2 = (00.0000, +1.0000, 00.0000)	2,722.6000	73.2059	11.1420	7.5893	359.8321
O_3 = (00.0000, 00.0000, −1.0000)	1,714.3000	66.5848	10.7888	5.6308	272.1722
O_4 = (00.0000, 00.0000, +1.0000)	1,897.5000	68.1690	10.7888	5.6502	273.2678
O_5 = (−0.6508, +0.7593, 00.0000)	2,256.8000	69.4838	10.6408	11.0232	510.5843
O_6 = (+0.6508, −0.7593, 00.0000)	2,095.1000	70.4126	10.6408	11.0061	509.6172
O_7 = (+0.8321, +0.5547, 00.0000)	1,362.8000	66.8424	10.3246	13.4461	616.2106
O_8 = (−0.8321, −0.5547, 00.0000)	953.4000	67.8157	10.3246	13.4028	613.7621

Compared to support volume, volumetric error, surface roughness, build time and build cost, the strength, elongation, hardness, residual stress, flexural modulus and fatigue performance of an LPBF part in a given build orientation are more difficult to estimate. Although several existing OBO selection methods, such as the methods of Zhang et al. (2016), Brika et al. (2017), Ransikarbum and Kim (2017), Cheng and To (2019) and Raju et al. (2019), have considered some of these attributes, the estimation models in them are either not for the LPBF process or incomplete. From other existing literature, e.g. the research papers of Bartolomeu et al. (2016), Miranda et al. (2016), Mukherjee et al. (2017), Williams et al. (2018), Li et al. (2018; 2019) and Ganeriwala et al. (2019), some related prediction models can be found. However, these prediction models cannot be used in the proposed method since the build orientation of an LPBF part is not a variable in them. To this end, the estimation of the strength, elongation, hardness, residual stress, flexural modulus and fatigue performance of an LPBF part in a given build orientation is not considered in the proposed method for the time being.

5.2 Normalisation of attribute values

For the transformation of the estimated attribute values into fuzzy numbers, a specific membership function is required since the values are generally in crisp numbers. Brauers et al. (2008) tested a set of ratio models and found that the best ratio model for converting crisp numbers into fuzzy numbers in MADM is the square root of the sum of squares of each option per attribute for such denominator. This model is used as the membership function to convert attribute values into fuzzy numbers to establish a fuzzy decision matrix in the proposed automatic selection method.

In general, there are two different types of attributes in an MADM problem, i.e. benefit and cost attributes. Benefit attributes have positive effect on the decision making result, while cost attributes affect the result adversely (Greco et al., 2016). For example, tensile strength and surface roughness respectively belong to benefit and cost attributes in selection of the OBO of an LPBF part, since they respectively have positive and negative influences on the selection result. To unify the effect of different types of attributes on the decision making result in fuzzy MADM problems, a complement rule is usually adopted to normalise the fuzzy numbers quantifying the values of cost attributes. In the proposed method, this rule is also used to normalise the established fuzzy decision matrix.

The basic components of an MADM problem include a set of options $\boldsymbol{O} = \{O_1, O_2, \ldots, O_m\}$, a set of attributes $\boldsymbol{A} = \{A_1, A_2, \ldots, A_n\}$, a vector of weights of attributes $\boldsymbol{w} = [w_1, w_2, \ldots, w_n]^{\mathrm{T}}$ and a decision matrix $\boldsymbol{M} = [v_{i,j}]_{m\times n}$ ($i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$), where $O_1, O_2, \ldots, O_m$ are m different options, $A_1, A_2, \ldots, A_n$ are n different attributes, $v_{i,j}$ is the value of the j-th attribute of the i-th option, and $w_1, w_2, \ldots, w_n$ are respectively the weights of $A_1, A_2, \ldots, A_n$ such that $0 \leq w_1, w_2, \ldots, w_n \leq 1$ and $w_1 + w_2 + \ldots + w_n = 1$. For an OBO selection problem, the generated ABOs, the attributes of ABOs, the relative importance of attributes and the estimated attribute values can be respectively regarded as the options, attributes, attribute weights and attribute values of an MADM problem. Based on this, the normalisation of attribute values is formally described as follows:

- Fuzzify the estimated attribute values. The estimated attribute values are converted into fuzzy numbers using the following ratio model:

$$r_{i,j} = v_{i,j} \Big/ \sqrt{\sum_{i=1}^{m} v_{i,j}^2} \tag{5.12}$$

  where $r_{i,j}$ is the ratio of $v_{i,j}$. It is not difficult to prove that $0 \leq r_{i,j} \leq 1$. Therefore, the crisp value $v_{i,j}$ can be fuzzified as the fuzzy number $<r_{i,j}>$. For example, the estimated attribute values in Table 5.4 are fuzzified as the fuzzy numbers listed in Table 5.5 using the ratio model.

Table 5.5 The fuzzified attribute values of the generated ABOs

ABO	$\boldsymbol{A_1}$	$\boldsymbol{A_2}$	$\boldsymbol{A_3}$	$\boldsymbol{A_4}$	$\boldsymbol{A_5}$
O_1	<0.7448>	<0.3559>	<0.3672>	<0.2818>	<0.2899>
O_2	<0.3550>	<0.3749>	<0.3672>	<0.2706>	<0.2762>
O_3	<0.2235>	<0.3410>	<0.3556>	<0.2007>	<0.2089>
O_4	<0.2474>	<0.3491>	<0.3556>	<0.2014>	<0.2097>
O_5	<0.2942>	<0.3559>	<0.3507>	<0.3930>	<0.3919>
O_6	<0.2731>	<0.3606>	<0.3507>	<0.3924>	<0.3911>
O_7	<0.1777>	<0.3423>	<0.3403>	<0.4794>	<0.4729>
O_8	<0.1243>	<0.3473>	<0.3403>	<0.4778>	<0.4710>

Notes: A_1: Support volume; A_2: Volumetric error; A_3: Surface roughness; A_4: Build time; A_5: Build cost

- Established a fuzzy decision matrix. According to the transformation result, a fuzzy decision matrix is established as:

$$\boldsymbol{M} = \begin{bmatrix} <r_{1,1}> & <r_{1,2}> & \cdots & <r_{1,n}> \\ <r_{2,1}> & <r_{2,2}> & \cdots & <r_{2,n}> \\ \vdots & \vdots & \ddots & \vdots \\ <r_{m,1}> & <r_{m,2}> & \cdots & <r_{m,n}> \end{bmatrix} \tag{5.13}$$

  For instance, a fuzzy decision matrix for the estimated attribute values in Table 5.4 is established as $\boldsymbol{M} = [<r_{i,j}>]_{8\times5}$, where the values of $<r_{i,j}>$ (i = 1, 2, …, 8; j = 1, 2, …, 5) are listed in Table 5.5.

- Normalise the fuzzy decision matrix. Using the complement rule, the fuzzy decision matrix $\boldsymbol{M}$ is normalised as:

$$\boldsymbol{N} = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,n} \\ \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{m,1} & \alpha_{m,2} & \cdots & \alpha_{m,n} \end{bmatrix} \tag{5.14}$$

  where $\alpha_{i,j} = <r_{i,j}>$ if A_j is a benefit attribute and $\alpha_{i,j} = <1 - r_{i,j}>$ if A_j is a cost attribute. As an example, for each of the attributes support volume, volumetric error, surface roughness, build time and build cost, the objective is to minimise the value of the attribute. Thus, each of them is a cost attribute. The fuzzy decision matrix $\boldsymbol{M} = [<r_{i,j}>]_{8\times5}$ can be normalised as $\boldsymbol{N} = [\alpha_{i,j}]_{8\times5}$, where the values of $\alpha_{i,j}$ are listed in Table 5.6.

5.3 Selection of the optimal orientation

The attributes of ABOs may be independent, but more often they are interrelated. The relationships of attributes can be determined according to the estimation models used or the research findings in the existing literature. For example, it can be determined that build cost and support

volume are interrelated for LPBF AM, since from the estimation model of build cost in Equation (5.8), material cost is a part of build cost and the larger the support volume, the higher the material cost. According to the used estimation models of the attributes support volume, volumetric error, surface roughness, build time and build cost, the relationships among them can be determined as listed in Table 5.7.

Table 5.6 The normalised attribute values of the generated ABOs

ABO	$\boldsymbol{A_1}$	$\boldsymbol{A_2}$	$\boldsymbol{A_3}$	$\boldsymbol{A_4}$	$\boldsymbol{A_5}$
O_1	<0.2552>	<0.6441>	<0.6328>	<0.7182>	<0.7101>
O_2	<0.6450>	<0.6251>	<0.6328>	<0.7294>	<0.7238>
O_3	<0.7765>	<0.6590>	<0.6444>	<0.7993>	<0.7911>
O_4	<0.7526>	<0.6509>	<0.6444>	<0.7986>	<0.7903>
O_5	<0.7058>	<0.6441>	<0.6493>	<0.6070>	<0.6081>
O_6	<0.7269>	<0.6394>	<0.6493>	<0.6076>	<0.6089>
O_7	<0.8223>	<0.6577>	<0.6597>	<0.5206>	<0.5271>
O_8	<0.8757>	<0.6527>	<0.6597>	<0.5222>	<0.5290>

Notes: A_1: Support volume; A_2: Volumetric error; A_3: Surface roughness; A_4: Build time; A_5: Build cost

Table 5.7 The relationships among the five considered attributes

ABO	**Support volume**	**Volumetric error**	**Surface roughness**	**Build time**	**Build cost**
Support volume	————	Independent	Independent	Interrelated	Interrelated
Volumetric error	Independent	————	Independent	Independent	Independent
Surface roughness	Independent	Independent	————	Independent	Independent
Build time	Interrelated	Independent	Independent	————	Interrelated
Build cost	Interrelated	Independent	Independent	Interrelated	————

In the proposed method, the weights of the considered attributes of ABOs are used to measure their relative importance for selection of the OBO. The values of attribute weights are either directly assigned by users or determined through pairwise comparison (Saaty, 1977) in the analytic hierarchy process.

Once the relationships and weights of the attributes of ABOs are determined, the normalised attribute values of each ABO can be aggregated into a single fuzzy number using the constructed fuzzy AOs. The aggregation mainly includes calculation of the power weight of each normalised attribute value and calculation of the aggregated attribute value of each ABO. According to the aggregation result, the ABOs can be ranked and then the OBO can be selected.

Formally, suppose O_1, O_2, …, O_m are m generated ABOs of an LPBF part, A_1, A_2, …, A_n are n considered attributes of ABOs and the fuzzy numbers $\alpha_{i,j}$ (i = 1, 2, …, m; j = 1, 2, …, n) in the matrix $\boldsymbol{N}$ in Equation (5.14) are the normalised attribute values of O_i with respect to A_j. Then the selection of the OBO for the part includes the following steps:

- Compute the values of the weights of A_j. If a user directly provides the values of the weights of attributes, this step will be skipped. If a user explicitly specifies the relative importance of each pair of attributes for selection of the OBO, the weight values will be calculated using a scaling approach based on pairwise comparison (Saaty, 1977) in the analytic hierarchy process. This approach determines the weight vector $\boldsymbol{w} = [w_1, w_2, \ldots, w_n]^T$ from a positive reciprocal pairwise comparison matrix:

$$\boldsymbol{C} = \begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_n \end{array} \begin{array}{c} \begin{array}{cccc} A_1 & A_2 & \cdots & A_n \end{array} \\ \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,n} \end{bmatrix} \end{array} \tag{5.15}$$

where $c_{i,j}$ is the scale value which stands for the degree of importance of the attribute A_i over the attribute A_j for selection of the OBO and are defined in Figure 5.4.

Scale ($c_{i,j}$)	Meaning of the scale value for selection of the optimal orientation
1	The attribute A_i and the attribute A_j are equally important
3	The attribute A_i is slightly more important than the attribute A_j
5	The attribute A_i is moderately more important than the attribute A_j
7	The attribute A_i is strongly more important than the attribute A_j
9	The attribute A_i is extremely more important than the attribute A_j
2, 4, 6, 8	Intermediate values for compromise of two degrees of importance
$1/c_{i,j}$	Degree of importance of the attribute A_j over the attribute A_i

Figure 5.4 Definition of the elements of a pairwise comparison matrix

According to the positive reciprocity, all $c_{i,j}$ must satisfy the conditions that $c_{i,j} > 0$ and $c_{i,j} = 1/c_{j,i}$. The matrix $\boldsymbol{C}$ is consistent if and only if $c_{i,k}c_{k,j} = c_{i,j}$ for all $i, j, k = 1, 2, \ldots, n$. In practical applications, the matrix is not necessarily consistent. But the inconsistencies must be controlled within a certain range. To define the range of inconsistencies, three indicators named consistency index (CI), random index (RI) and consistency ratio (CR) were introduced. The value of CI can be calculated via the following equation:

$$CI = \left(\lambda_{\max} - n\right)/(n-1) \tag{5.16}$$

where $\lambda_{\max}$ is the maximum eigenvalue of the matrix $\boldsymbol{C}$. RI is the average consistency rate of

500 randomly constructed n-order pairwise comparison matrices $\boldsymbol{C}_1$, $\boldsymbol{C}_2$, …, $\boldsymbol{C}_{500}$, whose value can be computed using the following equation:

$$RI = \left(CI_1 + CI_2 + \cdots + CI_{500}\right) / 500 \tag{5.17}$$

where CI_1, CI_2, …, CI_{500} are the CI values of the matrices $\boldsymbol{C}_1$, $\boldsymbol{C}_2$, …, $\boldsymbol{C}_{500}$, respectively. The values of RI from $n = 1$ to $n = 15$ are listed in Table 5.8.

Table 5.8 The random index values when the matrix order is within fifteen

n	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
RI	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.54	1.56	1.57	1.59

CR is defined as the ratio of CI and RI, whose value can be calculated through the following equation:

$$CR = CI / RI \tag{5.18}$$

If the CR value of a pairwise comparison matrix $\boldsymbol{C}$ is negative, then $\boldsymbol{C}$ is not a positive reciprocal matrix and the elements of $\boldsymbol{C}$ need to be adjusted until all $c_{i,j}$ satisfy the conditions that $c_{i,j} > 0$ and $c_{i,j} = 1/c_{j,i}$. If $CR = 0$, then $\boldsymbol{C}$ is a consistent matrix. If $0 < CR < 0.10$, then the inconsistencies in $\boldsymbol{C}$ are acceptable. When $\boldsymbol{C}$ is consistent or its inconsistencies are acceptable, the normalised principal eigenvector is used as the weight vector $\boldsymbol{w}$. That is, the normalised $\boldsymbol{w}'$ ($\boldsymbol{C}\boldsymbol{w}' = \lambda_{\max}\boldsymbol{w}'$) is used as $\boldsymbol{w}$. Otherwise, the inconsistent scale values in the matrix $\boldsymbol{C}$ need to be adjusted until $0 \le CR < 0.10$. To sum up, the determination of attribute weights using the scaling approach based on pairwise comparison includes three steps. The first step is to construct a positive reciprocal pairwise comparison matrix $\boldsymbol{C}$ according to Equation (5.15). The second step is to calculate the CR value of the matrix $\boldsymbol{C}$ using Equation (5.18). In the last step, the attribute weights are determined as the elements of the normalised principal eigenvector of the matrix $\boldsymbol{C}$ if $0 \le CR < 0.10$. Otherwise, the inconsistent scale values in the matrix $\boldsymbol{C}$ are adjusted and the second step is repeated until $0 \le CR < 0.10$. For example, suppose the degrees of importance of the attributes support volume, volumetric error, surface roughness, build time and build cost for selection of the OBO are expressed in the following positive reciprocal pairwise comparison matrix:

$$\boldsymbol{C} = \begin{bmatrix} 1 & 3 & 3 & 1/2 & 1/2 \\ 1/3 & 1 & 1 & 1/4 & 1/4 \\ 1/3 & 1 & 1 & 1/2 & 1/2 \\ 2 & 4 & 2 & 1 & 1/5 \\ 2 & 4 & 2 & 5 & 1 \end{bmatrix}$$

The maximum eigenvalue of this matrix is $\lambda_{max} = 5.4899$. According to Equation (5.18), the CR value of the matrix $\boldsymbol{C}$ is 0.1094. Since $CR > 0.10$, the inconsistencies in $\boldsymbol{C}$ are unacceptable. If the scale values $c_{4,5}$ and $c_{5,4}$ in $\boldsymbol{C}$ are respectively adjusted to 1/2 and 2, that is, the degrees of importance of the five attributes are expressed by the following matrix:

$$\boldsymbol{C} = \begin{bmatrix} 1 & 3 & 3 & 1/2 & 1/2 \\ 1/3 & 1 & 1 & 1/4 & 1/4 \\ 1/3 & 1 & 1 & 1/2 & 1/2 \\ 2 & 4 & 2 & 1 & 1/2 \\ 2 & 4 & 2 & 2 & 1 \end{bmatrix}$$

then $\lambda_{max} = 5.2097$ and $CR = 0.0468$. Since $0 < CR < 0.10$, the inconsistencies are acceptable. The principal eigenvector of $\boldsymbol{C}$ is $\boldsymbol{w'} = [0.4095, 0.1561, 0.2147, 0.5275, 0.6954]^T$. By normalising $\boldsymbol{w'}$, the attribute weight vector is $\boldsymbol{w} = [0.2044, 0.0779, 0.1072, 0.2633, 0.3472]^T$.

- Compute the values of the power weights of $\alpha_{i,j}$. Based on Equation (4.26), the power weight values of the normalised attribute values can be calculated using the following equation:

$$\varphi_{i,j} = \frac{w_j \left(1 + \sum_{\substack{j'=1 \\ j' \neq j}}^{n} \left(1 - d(\alpha_{i,j},\ \alpha_{i,j'}) \right) \right)}{\sum_{k=1}^{n} \left(w_k \left(1 + \sum_{\substack{k'=1 \\ k' \neq k}}^{n} \left(1 - d(\alpha_{i,k},\ \alpha_{i,k'}) \right) \right) \right)} \tag{5.19}$$

where $w_1, w_2, \ldots, w_n$ are respectively the assigned or determined weights of the attributes $A_1, A_2, \ldots, A_n$, $d(\alpha_{i,j}, \alpha_{i,j'})$ is the distance between $\alpha_{i,j}$ and $\alpha_{i,j'}$, and $d(\alpha_{i,k}, \alpha_{i,k'})$ is the distance between $\alpha_{i,k}$ and $\alpha_{i,k'}$. In this step, the Euclidean distance measure in Equation (4.2) is used to calculate the values of $d(\alpha_{i,j}, \alpha_{i,j'})$ and $d(\alpha_{i,k}, \alpha_{i,k'})$. For example, if the weights of support volume, volumetric error, surface roughness, build time and build cost are respectively 0.2044, 0.0779, 0.1072, 0.2633 and 0.3472, then using Equation (5.19), the power weight values of $\alpha_{i,j}$ ($i = 1, 2, \ldots, 8$; $j = 1, 2, \ldots, 5$) in Table 5.6 are calculated as follow:

$$\left[\varphi_{i,j} \right]_{8\times5} = \begin{bmatrix} 0.1622 & 0.2068 & 0.2062 & 0.2066 & 0.1995 & 0.1970 & 0.1902 & 0.1835 \\ 0.0832 & 0.0782 & 0.0766 & 0.0770 & 0.0789 & 0.0790 & 0.0812 & 0.0818 \\ 0.1141 & 0.1082 & 0.1045 & 0.1056 & 0.1084 & 0.1086 & 0.1117 & 0.1124 \\ 0.2754 & 0.2611 & 0.2635 & 0.2626 & 0.2644 & 0.2653 & 0.2654 & 0.2677 \\ 0.3651 & 0.3457 & 0.3492 & 0.3482 & 0.3488 & 0.3501 & 0.3515 & 0.3546 \end{bmatrix}^T$$

- Compute the aggregated attribute values of O_i. The aggregated attribute values of ABOs are calculated using the following equations:

$$\alpha_{\mathrm{mm},i} = FAWPMM^{Q}(\alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,n}) \tag{5.20}$$

$$\alpha_{\mathrm{gmm},i} = FAWPGMM^{Q}(\alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,n}) \tag{5.21}$$

where *FAWPMM* is a specific FAWPMM operator, such as the operators in Equation (4.31), Equation (4.32), Equation (4.34) and Equation (4.36) and *FAWPGMM* is a specific FAWPGMM operator, such as the operators in Equation (4.44), Equation (4.45), Equation (4.47) and Equation (4.49), and $Q = (q_1, q_2, \ldots, q_n)$ is used to capture the interrelationships among the considered attributes of ABOs in selection of the OBO: If the considered attributes are independent of each other, it can be assigned that $q_1 > 0$ and $q_2 = q_3 = \ldots = q_n = 0$; If there are interrelationships between two considered attributes, it can be assigned that $q_1, q_2 > 0$ and $q_3 = q_4 = \ldots = q_n = 0$; If there are interrelationships among k ($k = 3, 4, \ldots, n$) considered attributes, it can be assigned that $q_1, q_2, \ldots, q_k > 0$ and $q_{k+1} = q_{k+2} = \ldots = q_n = 0$. For instance, because there are interrelationships among support volume, build time and build cost according to Table 5.7, $Q = (q_1, q_2, q_3, q_4, q_5)$ for aggregating $\alpha_{i,j}$ ($i = 1, 2, \ldots, 8$; $j = 1, 2, \ldots, 5$) in Table 5.6 can be assigned as $Q = (1, 1, 1, 0, 0)$. The aggregated attribute values of the ABOs O_1, O_2, …, O_8 are calculated using the specific FAWPMM operator in Equation (4.34) and the specific FAWPGMM operator in Equation (4.47) with $\delta = 3$. The calculated results are listed in Table 5.9.

Table 5.9 The aggregated attribute values of the generated ABOs

O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8
$\alpha_{\mathrm{mm},1}$	$\alpha_{\mathrm{mm},2}$	$\alpha_{\mathrm{mm},3}$	$\alpha_{\mathrm{mm},4}$	$\alpha_{\mathrm{mm},5}$	$\alpha_{\mathrm{mm},6}$	$\alpha_{\mathrm{mm},7}$	$\alpha_{\mathrm{mm},8}$
<0.5528>	<0.6303>	<0.6804>	<0.6750>	<0.6038>	<0.6076>	<0.5981>	<0.6110>
$\alpha_{\mathrm{gmm},1}$	$\alpha_{\mathrm{gmm},2}$	$\alpha_{\mathrm{gmm},3}$	$\alpha_{\mathrm{gmm},4}$	$\alpha_{\mathrm{gmm},5}$	$\alpha_{\mathrm{gmm},6}$	$\alpha_{\mathrm{gmm},7}$	$\alpha_{\mathrm{gmm},8}$
<0.6207>	<0.6960>	<0.7681>	<0.7615>	<0.6489>	<0.6536>	<0.6369>	<0.6494>

- Compute the combined score values of the aggregated attribute values of O_i. The score values of the aggregated attribute values of Equation (5.20) and Equation (5.21) are calculated using Equation (4.1). Let $S(\alpha_{\mathrm{mm},i})$ and $S(\alpha_{\mathrm{gmm},i})$ be the calculated score values. To balance the opposite aggregation expectations generated by the FAWPMM and FAWPGMM operators, an average combination of their score values

$$S(\alpha_i) = \left(S(\alpha_{\mathrm{mm},i}) + S(\alpha_{\mathrm{gmm},i})\right)/2 \tag{5.22}$$

will be applied to rank the ABOs. As an example, the score values of the aggregated attributes in Table 5.9 are calculated by Equation (4.1). Then the combined score values are calculated using Equation (5.22). The calculated results are listed in Table 5.10.

Table 5.10 The calculated score values of the generated ABOs

ABO	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8
$S(\alpha_{mm,i})$	+0.1055	+0.2607	+0.3609	+0.3500	+0.2076	+0.2153	+0.1962	+0.2219
$S(\alpha_{gmm,i})$	+0.2414	+0.3920	+0.5361	+0.5231	+0.2979	+0.3072	+0.2738	+0.2989
$S(\alpha_i)$	+0.1735	+0.3264	+0.4485	+0.4366	+0.2527	+0.2613	+0.2350	+0.2604

- Generate a ranking of O_i. On the basis of the calculated combined score values, the generated ABOs are ranked according to the comparison rule in Definition 4.4. For example, according to the combined score values in Table 5.10, a ranking of the generated ABOs is obtained as follow:

$$O_3 \succ O_4 \succ O_2 \succ O_6 \succ O_8 \succ O_5 \succ O_7 \succ O_1$$

- Select the OBO from O_i. Each of the ABOs ranked first can be selected as the OBO of the part. For instance, according to the ranking, O_3 is the only ABO ranked first, it is selected as the OBO of the part in Figure 5.3. A brief summarisation of the build orientation determination results for the part is depicted in Figure 5.5.

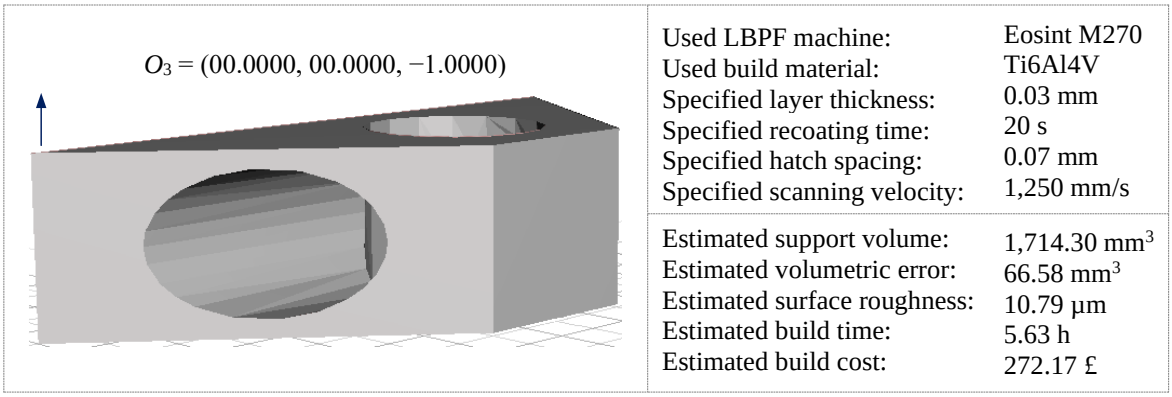


Figure 5.5 Build orientation determination results of an LPBF part

5.4 Summary

This chapter has presented an MADM-based method for automatic selection of the OBO for LPBF AM. This method includes estimation of attribute values, normalisation of attribute values and selection of the OBO. The estimation of attribute values takes: the STL model and ABOs of an LPBF part and the material, layer thickness, recoating time, hatch spacing and laser scanning velocity to build the part as input, and outputs the estimated values of the support volume, volumetric error, surface roughness, build time and build cost of the part in each ABO. In the normalisation of attribute values, a ratio model is used to fuzzify the estimated values and a fuzzy decision matrix is established and normalised. The selection of the OBO firstly determines the rela-

tionships among support volume, volumetric error, surface roughness, build time and build cost. Then a scaling approach based on pairwise comparison is applied to calculate the weight values of these attributes. Based on the determined relationships and calculated weight values, the constructed FAWPMM and FAWPGMM operators in Chapter 4 are used to aggregate the normalised attribute values of each ABO. All ABOs are ranked according to the aggregation results and the OBO of the part can be selected based on the ranking. The presented MADM method will be demonstrated in Chapter 6.

6 Implementation and demonstration

In this chapter, a method for automatic determination of part build orientation for LPBF AM is first developed via combining and implementing the proposed automatic generation method in Chapter 3 and the proposed automatic selection method in Chapter 5. Then a set of build orientation determination cases are presented to illustrate the application of the developed method. Finally, the effectiveness, efficiency and advantages of the method are evaluated through theoretical analysis, experimental analysis and comparisons.

The chapter is organised as follows. Section 6.1 describes the development of the automatic determination method. Illustrations of the application of the method are reported in Section 6.2. The evaluation of the method is documented in Section 6.3. Section 6.4 ends the chapter with a summary.

6.1 Automatic determination method

Based on the presented ABO generation method in Chapter 3 and the presented OBO selection method in Chapter 5, an automatic determination method of part build orientation for LPBF AM is developed. The availability of the source code of the developed automatic determination method is stated in Appendix E. The framework of the method is depicted in Figure 6.1.

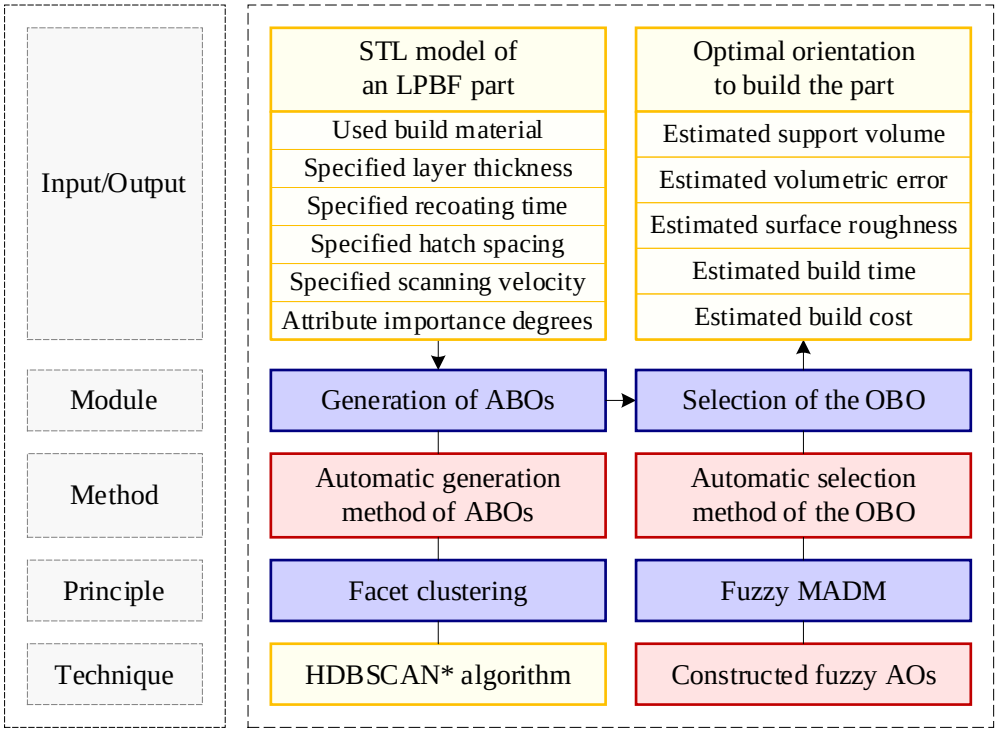


Figure 6.1 Framework of the automatic determination method

As can be seen from Figure 6.1, the developed method takes the STL model of an LPBF part, the material, layer thickness, recoating time, hatch spacing and scanning velocity for building the part and the degrees of importance of the attributes considered in build orientation determination as input, and outputs the optimal orientation to build the part and the estimated support volume, volumetric error, surface roughness, build time and build cost of the part in this orientation. It consists of an ABO generation module and an OBO selection module, which are developed via implementing the proposed novel facet clustering method for automatic generation of ABOs and the proposed novel MADM method for automatic selection of the OBO, respectively. Determination of the build orientation of an LPBF part using the developed method mainly includes the following steps:

- Import the STL model of the part, the material, layer thickness, recoating time, hatch spacing and scanning velocity to build the part and the degrees of importance of the attributes support volume, volumetric error, surface roughness, build time and build cost.
- Produce clusters of facets of the STL model. Using Rule 3.1, the accelerated HDBSCAN* algorithm and the *k*-cluster lifetime partition criterion, a certain number of meaningful clusters of facets are produced.
- Determine whether the produced clusters of facets need to be further refined and refine them when the answer is yes. If the number of the produced facet clusters for a regular (freeform) surface model exceeds 6 (12), these facet clusters will be refined using Rule 3.3. Otherwise, a further refinement is not required.
- Generate the ABOs of the part. According to Rule 3.2, the ABOs of each (refined) facet cluster are generated. The ABOs of the part are obtained by combining the generated ABOs of all facet clusters and removing the duplicated ABOs among them.
- Estimate the values of the attributes in each generated ABO. Using the estimation models in Section 5.1, the values of the support volume, volumetric error, surface roughness, build time and build cost in each generated ABO are estimated.
- Fuzzify the estimated attribute values. The estimated values of the support volume, volumetric error, surface roughness, build time and build cost in each generated ABO are converted into fuzzy numbers using the ratio model in Equation (5.12).
- Established a fuzzy decision matrix. According to the fuzzified result and Equation (5.13), a fuzzy decision matrix is established.
- Normalise the fuzzy decision matrix. According to Equation (5.14), the established fuzzy decision matrix is normalised.

- Calculate the weight values of attributes. If the input degrees of importance of support volume, volumetric error, surface roughness, build time and build cost are explicit weight values, this step is skipped. If the input degrees of importance of these attributes are in the form of the pairwise comparison matrix in Equation (5.15), the weight values of the considered attributes will be calculated using the scaling approach described in Section 5.3.
- Calculate the power weight value of each normalised attribute value. Using Equation (5.19), the value of the power weight of each normalised attribute value is calculated.
- Calculate the aggregated attribute value of each generated ABO. Using Equation (5.20) and Equation (5.21), the aggregated value of the attributes of each generated ABO is calculated.
- Calculate the combined score value of each generated ABO. Using Equation (4.1) and Equation (5.22), the combined score value of each generated ABO is calculated.
- Obtain a ranking of all generated ABOs. According to the calculated combined score values and the comparison rule in Definition 4.4, a ranking of all generated ABOs is generated.
- Select the OBO for the part. Each of the first ranked ABOs can be selected as the OBO for the part.

6.2 Case studies

Twelve parts are used as cases to illustrate the application of the developed automatic determination method. The STL models of these parts are shown in Figure 6.2. The basic information of the parts is listed in Table 6.1. Among all STL models in Figure 6.2, the STL model of Part 5 was provided by Zhang et al. (2016). The STL models of the remaining parts were downloaded from Thingiverse, a free and open-source online community for sharing 3D models. The availability of these STL models is stated in Appendix E. Based on the surface types, the twelve STL models can be divided into two groups. The first group includes the STL models from Part 1 to Part 8, which are all regular surface models. The STL models from Part 9 to Part 12 constitute the second group, as all of them belong to freeform surface models.

Assume the twelve parts will be built using Ti6Al4V and the LPBF machine EOSINT M270 with the layer thickness, recoating time, hatch spacing and laser scanning velocity listed in Table 5.3. The material density, material porosity, material price and energy price have also been provided in Table 5.3. Before building the parts, the four preparation steps, build orientation determination, support generation, 3D model slicing and path planning, are needed to be completed in sequence. The present section focuses on the completion of build orientation determination using the developed automatic determination method. It is further assumed that support volume, volu-

metric error, surface roughness, build time and build cost are the attributes considered in the determination of the build orientation of each part and the degrees of importance of these attributes are specified in the following positive reciprocal pairwise comparison matrix:

$$\boldsymbol{C} = \begin{bmatrix} 1 & 2 & 2 & 1 & 1 \\ 1/2 & 1 & 2 & 1/2 & 1/2 \\ 1/2 & 1/2 & 1 & 1/3 & 1/3 \\ 1 & 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 & 1 \end{bmatrix}$$

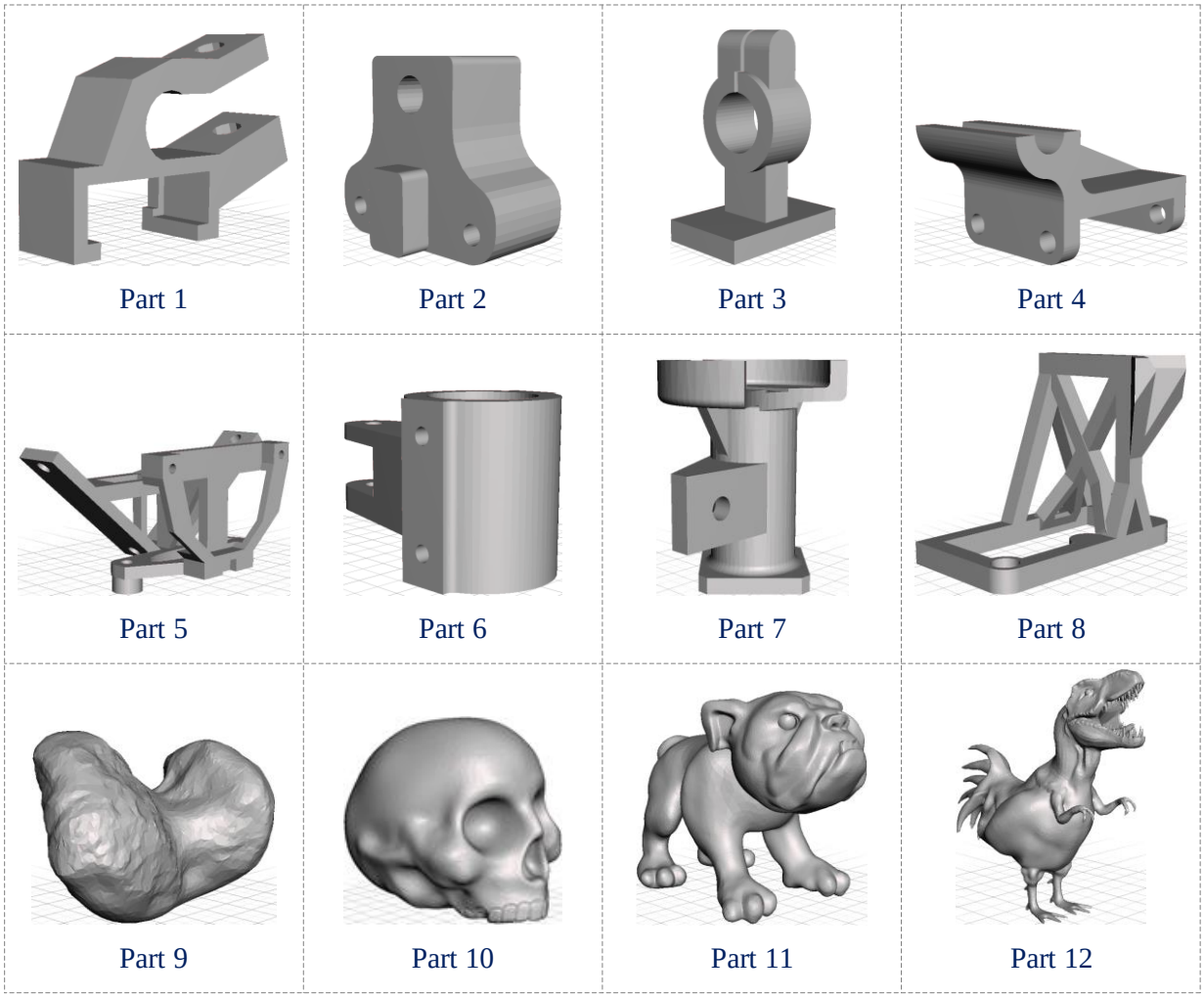


Figure 6.2 The STL models of twelve LPBF parts

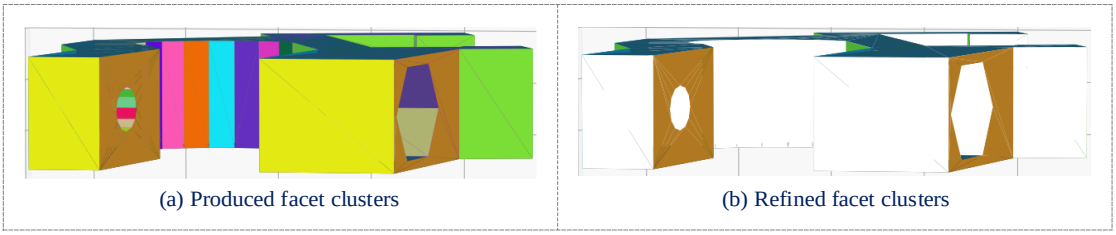
Based on the conditions above, the build orientation for each of the twelve parts can be automatically determined using the developed method. For example, the build orientation of Part 1 is determined via the following steps:

- Import the STL model of Part 1 (Part 1.stl) and the material (Ti6Al4V), layer thickness (0.03 mm), recoating time (20 s), hatch spacing (0.07 mm) and scanning velocity (1,250 mm/s) to build Part 1 into the developed method.

Table 6.1 The basic information of the twelve LPBF parts

Part	**Vertices**	**Facets**	**Surface**	**Length** (unit: mm)	**Width** (unit: mm)	**Height** (unit: mm)	**Area** (unit: mm^2)	**Volume** (unit: mm^3)
Part 1	140	284	Regular	38.31	10.00	25.40	2,629.50	2,864.45
Part 2	410	828	Regular	27.00	16.00	23.00	2,535.94	4,593.24
Part 3	798	1,592	Regular	120.00	80.00	167.89	60,941.50	404,405.00
Part 4	1,005	2,022	Regular	138.00	82.00	70.00	46,114.40	239,056.00
Part 5	1,181	2,426	Regular	74.09	33.75	34.34	9,137.33	9,603.15
Part 6	1,876	3,776	Regular	125.00	75.00	80.00	46,576.40	228,889.00
Part 7	3,160	6,324	Regular	160.00	122.00	162.00	96,711.40	639,850.00
Part 8	3,167	6,598	Regular	110.00	78.23	90.00	30,577.60	67,915.80
Part 9	6,610	13,240	Freeform	35.89	26.48	22.04	2,408.50	7,354.99
Part 10	33,863	67,722	Freeform	119.81	88.13	94.90	36,451.00	535,216.00
Part 11	44,368	88,724	Freeform	86.06	56.88	68.00	17,165.80	88,637.20
Part 12	54,314	108,888	Freeform	132.16	58.46	125.00	23,930.60	120,669.00

- Produce clusters of facets of the STL model. Using Rule 3.1, the accelerated HDBSCAN* algorithm and the *k*-cluster lifetime partition criterion, thirty-seven clusters of facets are produced for Part 1. The produced facet clusters are depicted in Figure 6.3(a).
- Determine whether the produced clusters of facets need to be further refined and refine them when the answer is yes. Since the STL model of Part 1 is a regular surface model consisting of less than 120,000 facets and the number of the produced facet clusters exceeds 6, the produced facet clusters are refined using Rule 3.3. The refined facet clusters are depicted in Figure 6.3(b) (Please note that the remaining facets are displayed in white).



(a) Produced facet clusters (b) Refined facet clusters

Figure 6.3 The produced and refined facet clusters for Part 1

- Generate the ABOs of the part. According to Rule 3.2, the ABOs of each refined facet cluster are generated. Through combining the generated ABOs of all facet clusters and removing the duplicated ABOs among them, the ABOs of Part 1 are obtained as follows:

O_1 = (00.0000, 00.0000, −1.0000)

O_2 = (00.0000, 00.0000, +1.0000)

O_3 = (−0.3420, −0.9397, 00.0000)

O_4 = (+0.3420, +0.9397, 00.0000)

O_5 = (00.0000, +1.0000, 00.0000)

O_6 = (00.0000, −1.0000, 00.0000)

The schematic diagram of these ABOs is shown in Figure 6.4.

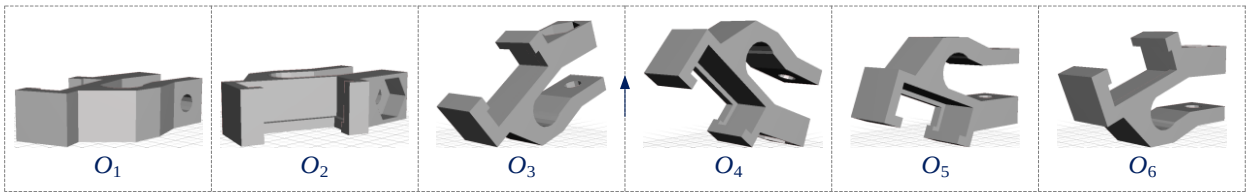


Figure 6.4 Schematic diagram of the generated ABOs of Part 1

- Estimate the values of the attributes in each generated ABO. Using the estimation models in Section 5.1, the values of the support volume, volumetric error, surface roughness, build time and build cost in each generated ABO of Part 1 are estimated. The estimated results are listed in Table 6.2.

Table 6.2 The estimated attribute values in the generated ABOs of Part 1

The generated ABO in unit vector	**Support volume** (unit: mm^3)	**Volumetric error** (unit: mm^3)	**Surface roughness** (unit: μm)	**Build time** (unit: h)	**Build cost** (unit: £)
O_1 = (00.0000, 00.0000, −1.0000)	218.6600	22.1375	10.4642	2.1781	100.0687
O_2 = (00.0000, 00.0000, +1.0000)	145.9500	22.1375	10.4642	2.1704	99.6339
O_3 = (−0.3420, −0.9397, 00.0000)	1,880.5100	24.8010	10.9248	5.6166	253.8252
O_4 = (+0.3420, +0.9397, 00.0000)	2,610.2700	23.8270	10.9248	5.6938	258.1897
O_5 = (00.0000, +1.0000, 00.0000)	2,408.1200	25.0293	10.9584	5.0262	228.4960
O_6 = (00.0000, −1.0000, 00.0000)	1,731.7600	25.2946	10.9584	4.9547	224.4509

- Fuzzify the estimated attribute values. The estimated values of the support volume, volumetric error, surface roughness, build time and build cost in each generated ABO of Part 1 are converted into fuzzy numbers using the ratio model in Equation (5.12). The converted results are listed in Table 6.3.
- Established a fuzzy decision matrix. According to the fuzzified result in Table 6.3 and Equation (5.13), a fuzzy decision matrix is established as $\boldsymbol{M} = [<r_{i,j}>]_{6\times5}$, where the values of $<r_{i,j}>$ (i = 1, 2, …, 6; j = 1, 2, …, 5) are listed in Table 6.3.

Table 6.3 The fuzzified attribute values of the generated ABOs of Part 1

ABO	A_1	A_2	A_3	A_4	A_5
O_1	<0.0499>	<0.3780>	<0.3961>	<0.1962>	<0.1987>
O_2	<0.0333>	<0.3780>	<0.3961>	<0.1955>	<0.1978>
O_3	<0.4290>	<0.4235>	<0.4135>	<0.5060>	<0.5040>
O_4	<0.5954>	<0.4069>	<0.4135>	<0.5129>	<0.5127>
O_5	<0.5493>	<0.4274>	<0.4148>	<0.4528>	<0.4537>
O_6	<0.3950>	<0.4319>	<0.4148>	<0.4463>	<0.4457>

Notes: A_1: Support volume; A_2: Volumetric error; A_3: Surface roughness; A_4: Build time; A_5: Build cost

- Normalise the fuzzy decision matrix. According to Equation (5.14), the established fuzzy decision matrix $\boldsymbol{M}$ is normalised as $\boldsymbol{N} = [\alpha_{i,j}]_{6\times5}$, where the values of $\alpha_{i,j}$ (i = 1, 2, …, 6; j = 1, 2, …, 5) are listed in Table 6.4.

Table 6.4 The normalised attribute values of the generated ABOs of Part 1

ABO	A_1	A_2	A_3	A_4	A_5
O_1	<0.9501>	<0.6220>	<0.6039>	<0.8038>	<0.8013>
O_2	<0.9667>	<0.6220>	<0.6039>	<0.8045>	<0.8022>
O_3	<0.5710>	<0.5765>	<0.5865>	<0.4940>	<0.4960>
O_4	<0.4046>	<0.5931>	<0.5865>	<0.4871>	<0.4873>
O_5	<0.4507>	<0.5726>	<0.5852>	<0.5472>	<0.5463>
O_6	<0.6050>	<0.5681>	<0.5852>	<0.5537>	<0.5543>

Notes: A_1: Support volume; A_2: Volumetric error; A_3: Surface roughness; A_4: Build time; A_5: Build cost

- Calculate the weight values of attributes. The weight values of the considered attributes are calculated by the scaling approach in Section 5.3: The maximum eigenvalue of the pairwise comparison matrix $\boldsymbol{C}$ is λ_{max} = 5.0394. According to Equation (5.18), the CR value of $\boldsymbol{C}$ is CR = 0.0088. Since 0 < CR < 0.10, the inconsistencies in $\boldsymbol{C}$ are acceptable. The principal eigenvector of $\boldsymbol{C}$ is $\boldsymbol{w'}$ = [0.5148, 0.2954, 0.1915, 0.5528, 0.5528]^T. By normalising $\boldsymbol{w'}$, the attribute weight vector is $\boldsymbol{w}$ = [0.2443, 0.1402, 0.0909, 0.2623, 0.2623]^T.
- Calculate the power weight value of each normalised attribute value. Using Equation (5.19), the power weight values of the normalised attribute values of the generated ABOs of Part 1 in Table 6.4 are calculated as follow:

$$\left[\varphi_{i,j}\right]_{6\times5} = \begin{bmatrix} 0.2281 & 0.2259 & 0.2466 & 0.2358 & 0.2336 & 0.2419 \\ 0.1394 & 0.1398 & 0.1414 & 0.1392 & 0.1418 & 0.1410 \\ 0.0892 & 0.0895 & 0.0911 & 0.0906 & 0.0912 & 0.0911 \\ 0.2716 & 0.2724 & 0.2603 & 0.2672 & 0.2667 & 0.2630 \\ 0.2717 & 0.2724 & 0.2606 & 0.2672 & 0.2667 & 0.2630 \end{bmatrix}^T$$

- Calculate the aggregated attribute value of each generated ABO. Using Equation (5.20) and Equation (5.21), the aggregated attribute value of each generated ABO of Part 1 is calculated. The calculated results are listed in Table 6.5.

Table 6.5 The aggregated attribute values of the generated ABOs of Part 1

O_1	O_2	O_3	O_4	O_5	O_6
$\alpha_{mm,1}$	$\alpha_{mm,2}$	$\alpha_{mm,3}$	$\alpha_{mm,4}$	$\alpha_{mm,5}$	$\alpha_{mm,6}$
<0.7350>	<0.7389>	<0.5272>	<0.4881>	<0.5231>	<0.5597>
$\alpha_{gmm,1}$	$\alpha_{gmm,2}$	$\alpha_{gmm,3}$	$\alpha_{gmm,4}$	$\alpha_{gmm,5}$	$\alpha_{gmm,6}$
<0.7980>	<0.8028>	<0.5421>	<0.5031>	<0.5387>	<0.5768>

- Calculate the combined score value of each generated ABO. Using Equation (4.1) and Equation (5.22), the combined score value of each generated ABO of Part 1 is calculated. The calculated results are listed in Table 6.6.

Table 6.6 The calculated score values of the generated ABOs of Part 1

ABO	O_1	O_2	O_3	O_4	O_5	O_6
$S(\alpha_{mm,i})$	+0.4701	+0.4778	+0.0543	−0.0238	+0.0462	+0.1193
$S(\alpha_{gmm,i})$	+0.5959	+0.6056	+0.0841	+0.0063	+0.0774	+0.1536
$S(\alpha_i)$	+0.5330	+0.5417	+0.0692	−0.0088	+0.0618	+0.1364

- Obtain a ranking of all generated ABOs. According to the comparison rule in Definition 4.4, a ranking of all generated ABOs of Part 1 is obtained as follow:

$$O_2 \succ O_1 \succ O_6 \succ O_3 \succ O_5 \succ O_4$$

- Select the OBO for the part. The ABOs ranked first are selected as the OBO for the part. As O_2 is the only ABO ranked first, it is selected as the OBO of Part 1. A brief summarisation of the build orientation determination results for Part 1 is depicted in Figure 6.5.

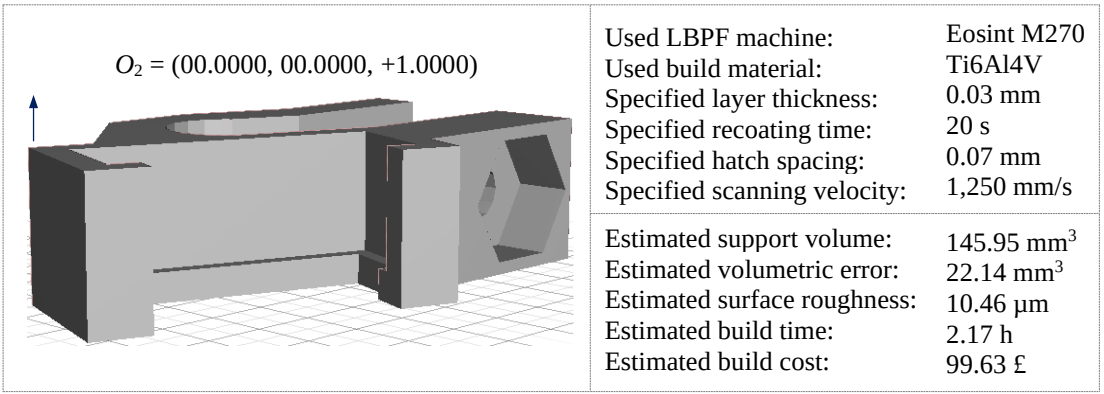


Figure 6.5 Build orientation determination results of Part 1

For each of the remaining eleven parts, the build orientation can be determined via the same steps. The results of facet clustering, ABO generation and OBO selection for these parts are directly provided in Figure 6.6 to Figure 6.27. It should be mentioned that the twenty-two figures does not include the results of facet clustering and ABO generation for Part 9, as they have been provided in Figure 3.7, Figure 3.11 and Figure 3.12. For the acquisition of the intermediate data in the process of build orientation determination of the eleven parts, please refer to Appendix E.

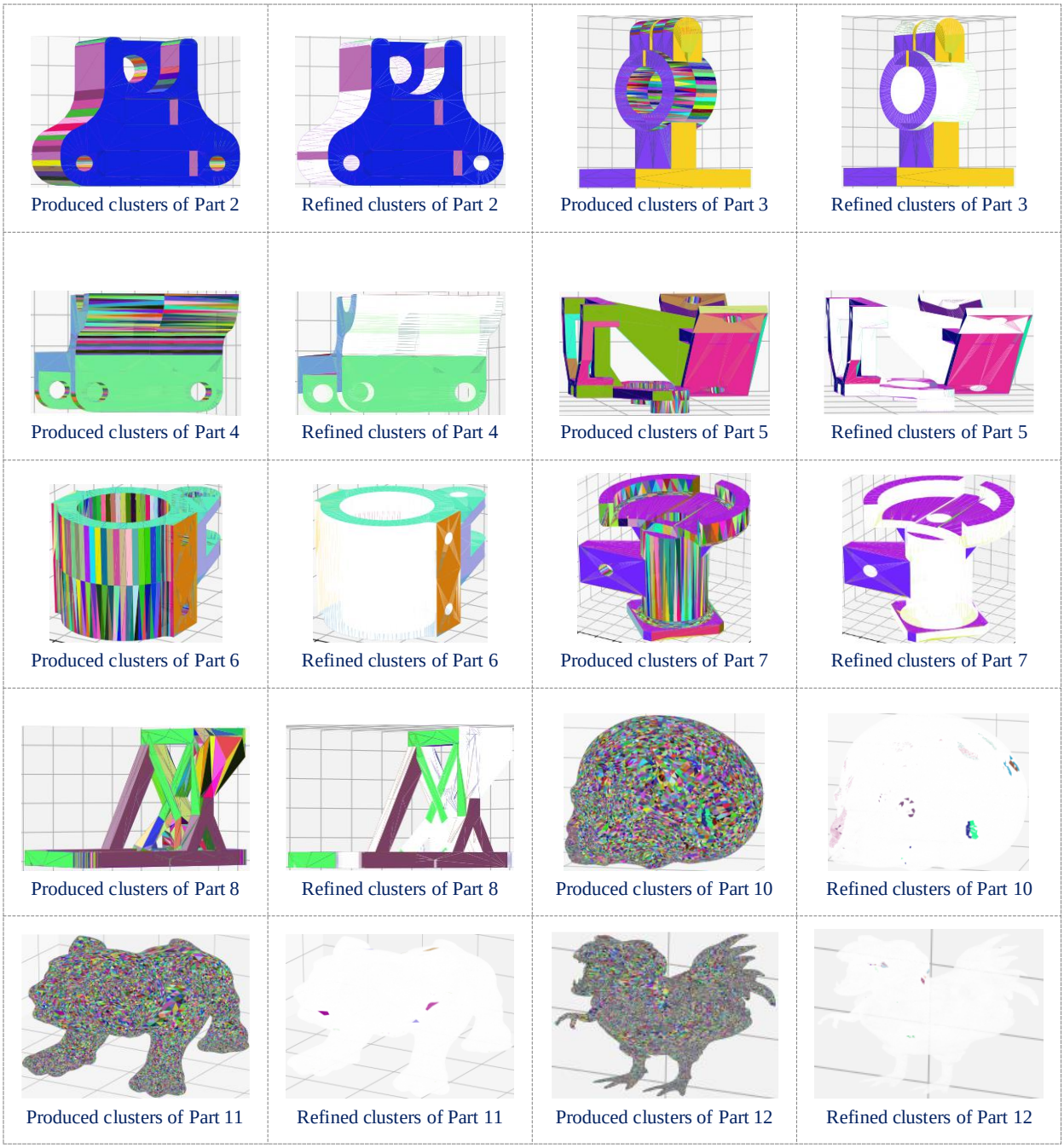


Figure 6.6 The produced and refined facet clusters of the remaining parts

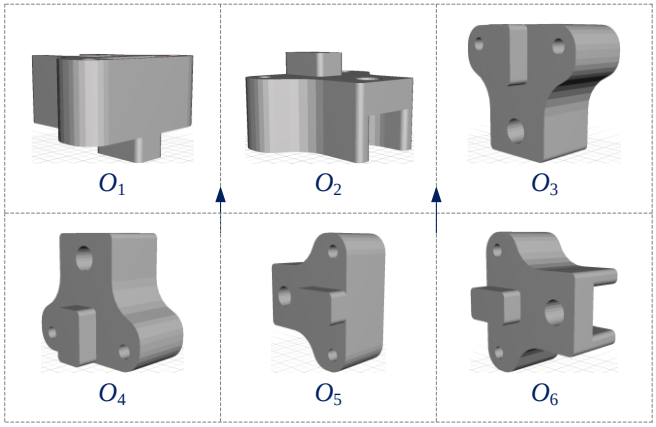
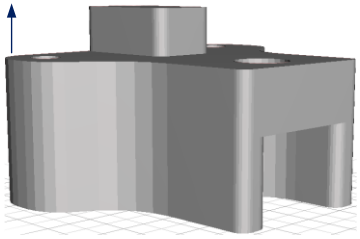


Figure 6.7 Schematic diagram of the generated ABOs of Part 2

$O_2 \succ O_4 \succ O_1 \succ O_3 \succ O_6 \succ O_5$

O_2 = (+1.0000, 00.0000, 00.0000)



Used LBPF machine:	Eosint M270
Used build material:	Ti6Al4V
Specified layer thickness:	0.03 mm
Specified recoating time:	20 s
Specified hatch spacing:	0.07 mm
Specified scanning velocity:	1,250 mm/s
Estimated support volume:	167.41 mm^3
Estimated volumetric error:	21.41 mm^3
Estimated surface roughness:	10.56 µm
Estimated build time:	3.47 h
Estimated build cost:	159.08 £

Figure 6.8 Build orientation determination results of Part 2

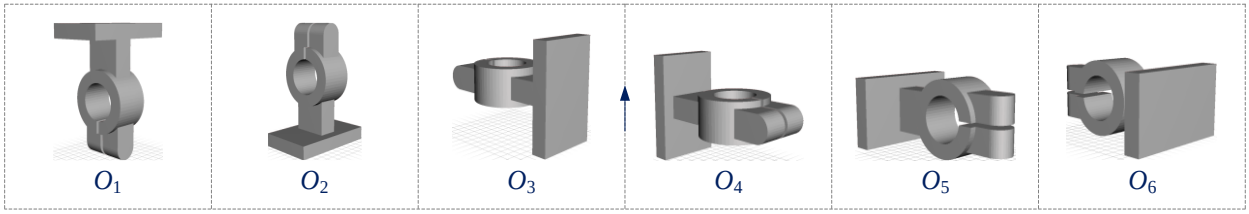
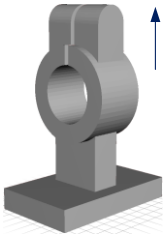


Figure 6.9 Schematic diagram of the generated ABOs of Part 3

$O_2 \succ O_4 \succ O_3 \succ O_5 \succ O_6 \succ O_1$

O_2 = (00.0000, 00.0000, +1.0000)



Used LBPF machine:	Eosint M270
Used build material:	Ti6Al4V
Specified layer thickness:	0.03 mm
Specified recoating time:	20 s
Specified hatch spacing:	0.07 mm
Specified scanning velocity:	1,250 mm/s
Estimated support volume:	13,377.00 mm^3
Estimated volumetric error:	551.77 mm^3
Estimated surface roughness:	10.80 µm
Estimated build time:	75.30 h
Estimated build cost:	3,869.09 £

Figure 6.10 Build orientation determination results of Part 3

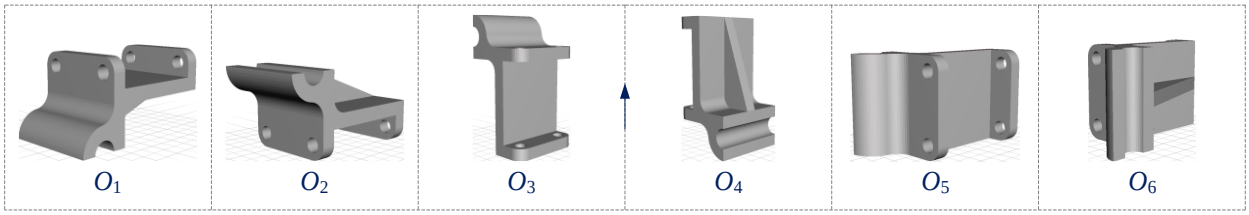
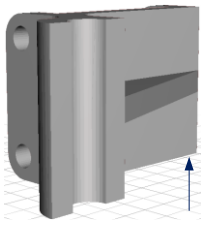


Figure 6.11 Schematic diagram of the generated ABOs of Part 4

$O_6 \succ O_5 \succ O_3 \succ O_2 \succ O_1 \succ O_4$

O_6 = (00.0000, +1.0000, 00.0000)



Used LBPF machine:	Eosint M270
Used build material:	Ti6Al4V
Specified layer thickness:	0.03 mm
Specified recoating time:	20 s
Specified hatch spacing:	0.07 mm
Specified scanning velocity:	1,250 mm/s
Estimated support volume:	6,114.00 mm^3
Estimated volumetric error:	360.61 mm^3
Estimated surface roughness:	10.13 µm
Estimated build time:	41.13 h
Estimated build cost:	2,135.65 £

Figure 6.12 Build orientation determination results of Part 4

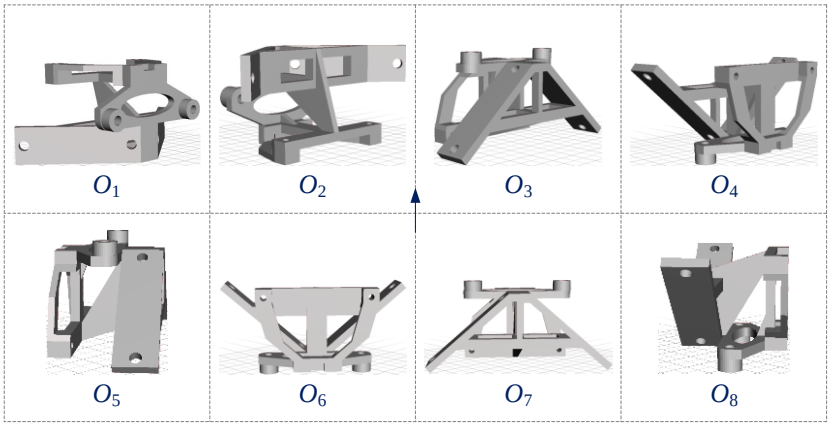
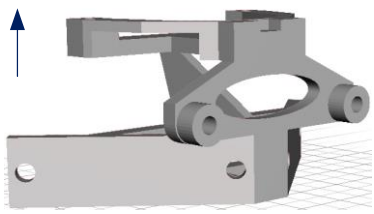


Figure 6.13 Schematic diagram of the generated ABOs of Part 5

$O_1 \succ O_2 \succ O_5 \succ O_7 \succ O_3 \succ O_8 \succ O_6 \succ O_4$

O_1 = (00.0000, +1.0000, 00.0000)



Used LBPF machine:	Eosint M270
Used build material:	Ti6Al4V
Specified layer thickness:	0.03 mm
Specified recoating time:	20 s
Specified hatch spacing:	0.07 mm
Specified scanning velocity:	1,250 mm/s
Estimated support volume:	4,587.15 mm^3
Estimated volumetric error:	78.85 mm^3
Estimated surface roughness:	10.61 µm
Estimated build time:	7.75 h
Estimated build cost:	360.37 £

Figure 6.14 Build orientation determination results of Part 5

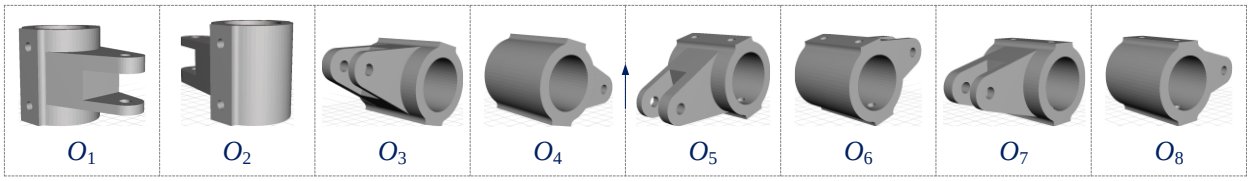
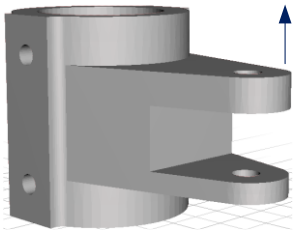


Figure 6.15 Schematic diagram of the generated ABOs of Part 6

$O_1 \succ O_2 \succ O_7 \succ O_8 \succ O_4 \succ O_5 \succ O_6 \succ O_3$

O_1 = (00.0000, 00.0000, −1.0000)



Used LBPF machine:	Eosint M270
Used build material:	Ti6Al4V
Specified layer thickness:	0.03 mm
Specified recoating time:	20 s
Specified hatch spacing:	0.07 mm
Specified scanning velocity:	1,250 mm/s
Estimated support volume:	13,022.00 mm^3
Estimated volumetric error:	375.12 mm^3
Estimated surface roughness:	10.29 µm
Estimated build time:	40.41 h
Estimated build cost:	2,099.83 £

Figure 6.16 Build orientation determination results of Part 6

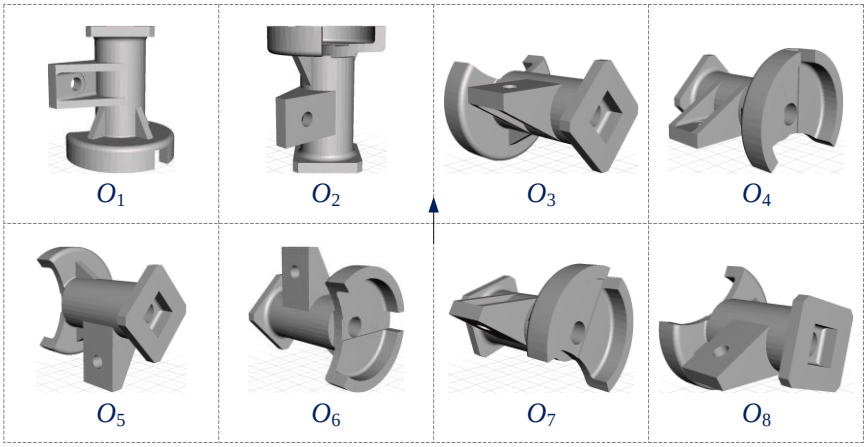
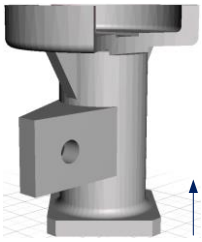


Figure 6.17 Schematic diagram of the generated ABOs of Part 7

$O_2 \succ O_1 \succ O_4 \succ O_5 \succ O_8 \succ O_3 \succ O_6 \succ O_7$

O_2 = (00.0000, 00.0000, −1.0000)



Used LBPF machine:	Eosint M270
Used build material:	Ti6Al4V
Specified layer thickness:	0.03 mm
Specified recoating time:	20 s
Specified hatch spacing:	0.07 mm
Specified scanning velocity:	1,250 mm/s
Estimated support volume:	39,032.00 mm^3
Estimated volumetric error:	840.03 mm^3
Estimated surface roughness:	10.66 µm
Estimated build time:	101.84 h
Estimated build cost:	5,382.57 £

Figure 6.18 Build orientation determination results of Part 7

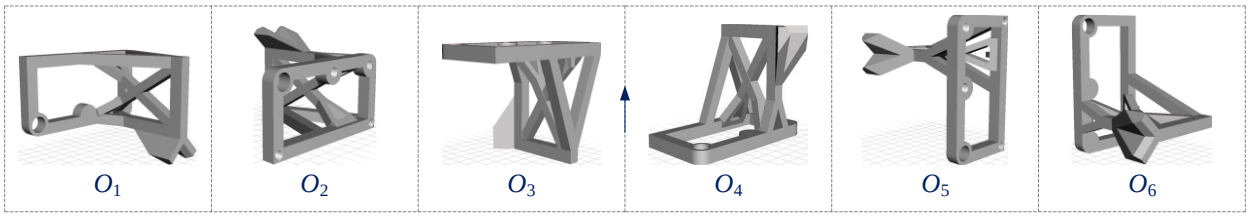
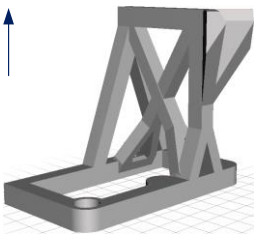


Figure 6.19 Schematic diagram of the generated ABOs of Part 8

$O_4 \succ O_3 \succ O_2 \succ O_1 \succ O_6 \succ O_5$

O_4 = (00.0000, 00.0000, −1.0000)

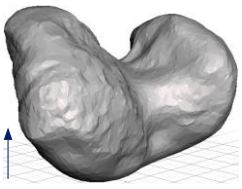


Used LBPF machine:	Eosint M270
Used build material:	Ti6Al4V
Specified layer thickness:	0.03 mm
Specified recoating time:	20 s
Specified hatch spacing:	0.07 mm
Specified scanning velocity:	1,250 mm/s
Estimated support volume:	3,060.60 mm^3
Estimated volumetric error:	273.39 mm^3
Estimated surface roughness:	10.46 μm
Estimated build time:	24.18 h
Estimated build cost:	1,159.14 £

Figure 6.20 Build orientation determination results of Part 8

$O_4 \succ O_{16} \succ O_{21} \succ O_{15} \succ O_{18} \succ O_7 \succ$
$O_8 \succ O_{12} \succ O_{11} \succ O_3 \succ O_{17} \succ O_6 \succ$
$O_{22} \succ O_{14} \succ O_2 \succ O_{20} \succ O_5 \succ O_{10} \succ$
$O_9 \succ O_{19} \succ O_1 \succ O_{13} \succ O_{23} \succ O_{24}$

O_4 = (−0.0007, +0.0001, +0.9999)



Used LBPF machine:	Eosint M270
Used build material:	Ti6Al4V
Specified layer thickness:	0.03 mm
Specified recoating time:	20 s
Specified hatch spacing:	0.07 mm
Specified scanning velocity:	1,250 mm/s
Estimated support volume:	1,183.37 mm^3
Estimated volumetric error:	22.87 mm^3
Estimated surface roughness:	10.70 μm
Estimated build time:	4.98 h
Estimated build cost:	230.95 £

Figure 6.21 Build orientation determination results of Part 9

It is worth noting that the determined OBO of each of the twelve parts above is the result under the attribute weight vector $\boldsymbol{w} = [0.2443, 0.1402, 0.0909, 0.2623, 0.2623]^T$. When the weight values of attributes change, the determined OBO could change accordingly. To illustrate the influence of weight values of attributes on the determined OBO, a sensitivity analysis experiment was carried out. In this experiment, the attribute weight vector $\boldsymbol{w}$ for each of the twelve parts was replaced by the following attribute weight vectors in turn:

$\boldsymbol{w}_1 = [0.8000, 0.0500, 0.0500, 0.0500, 0.0500]^T$

$\boldsymbol{w}_2 = [0.0500, 0.8000, 0.0500, 0.0500, 0.0500]^T$

$\boldsymbol{w}_3 = [0.0500, 0.0500, 0.8000, 0.0500, 0.0500]^T$

$\boldsymbol{w_4}$ = [0.0500, 0.0500, 0.0500, 0.8000, 0.0500]^T

$\boldsymbol{w_5}$ = [0.0500, 0.0500, 0.0500, 0.0500, 0.8000]^T

$\boldsymbol{w_6}$ = [0.3000, 0.0500, 0.0500, 0.3000, 0.3000]^T

$\boldsymbol{w_7}$ = [0.2375, 0.2375, 0.0500, 0.2375, 0.2375]^T

$\boldsymbol{w_8}$ = [0.2375, 0.0500, 0.2375, 0.2375, 0.2375]^T

$\boldsymbol{w_9}$ = [0.0500, 0.4250, 0.4250, 0.0500, 0.0500]^T

$\boldsymbol{w_{10}}$ = [0.2000, 0.2000, 0.2000, 0.2000, 0.2000]^T

where: Each of the five attributes (i.e. support volume, volumetric error, surface roughness, build time and build cost) is the most important attribute in turn from $\boldsymbol{w_1}$ to $\boldsymbol{w_5}$; From $\boldsymbol{w_6}$ to $\boldsymbol{w_9}$, the five attributes are divided into three groups of {support volume, build time, build cost}, {volumetric error} and {surface roughness} according to the determined relationships among them (see Table 5.7); In $\boldsymbol{w_6}$, $\boldsymbol{w_2}$, and $\boldsymbol{w_3}$, the attributes (attribute) in each group are (is) the most important attributes (attribute) in turn. The attributes in any two groups are the most important attributes from $\boldsymbol{w_7}$ to $\boldsymbol{w_9}$; In $\boldsymbol{w_{10}}$, the five attributes are equally important. The results of the experiment are the determined OBOs of the twelve parts under each attribute weight vector, which are listed in Table 6.7. It can be seen from the table that the determined OBOs of most parts (except Part 3 and Part 11) remain unchanged under all attribute weight vectors. This indicates that the attribute weights lose their due role for these parts.

Table 6.7 The determined OBOs of the twelve LPBF parts under each weight vector

Part	$\boldsymbol{w_1}$	$\boldsymbol{w_2}$	$\boldsymbol{w_3}$	$\boldsymbol{w_4}$	$\boldsymbol{w_5}$	$\boldsymbol{w_6}$	$\boldsymbol{w_7}$	$\boldsymbol{w_8}$	$\boldsymbol{w_9}$	$\boldsymbol{w_{10}}$
Part 1	O_2	O_2	O_2	O_2	O_2	O_2	O_2	O_2	O_2	O_2
Part 2	O_2	O_2	O_2	O_2	O_2	O_2	O_2	O_2	O_2	O_2
Part 3	O_2	O_2	O_4	O_2	O_2	O_2	O_4	O_2	O_4	O_4
Part 4	O_6	O_6	O_6	O_6	O_6	O_6	O_6	O_6	O_6	O_6
Part 5	O_1	O_1	O_1	O_1	O_1	O_1	O_1	O_1	O_1	O_1
Part 6	O_1	O_1	O_1	O_1	O_1	O_1	O_1	O_1	O_1	O_1
Part 7	O_2	O_2	O_2	O_2	O_2	O_2	O_2	O_2	O_2	O_2
Part 8	O_4	O_4	O_4	O_4	O_4	O_4	O_4	O_4	O_4	O_4
Part 9	O_4	O_4	O_4	O_4	O_4	O_4	O_4	O_4	O_4	O_4
Part 10	O_9	O_9	O_9	O_9	O_9	O_9	O_9	O_9	O_9	O_9
Part 11	O_{21}	O_{17}	O_{17}	O_{17}	O_{17}	O_{17}	O_{17}	O_{17}	O_{17}	O_{17}
Part 12	O_1	O_1	O_1	O_1	O_1	O_1	O_1	O_1	O_1	O_1

The reasons for such abnormal results can be explained from two aspects. On the one hand, the estimated support volume, build time and build cost are positively correlated on the basis of their estimation models. They tend to achieve the minimum in the same ABO. For example, the estimated support volume, build time and build cost in the six generated ABOs of Part 1 obtain

the minimum in O_2 (see Table 6.2). In this case, the changes in the weight values of the three attributes will not have any influence on the determined OBO. In other words, the three attributes could be reduced to one attribute, i.e. build cost. On the other hand, the influence of build orientation on the volumetric error and surface roughness of an LPBF part will be very slight or even negligible if the part is built by thin layers (Calignano, 2018). For instance, the difference among the estimated volumetric error or surface roughness in the six generated ABOs of Part 1 is quite small (see Table 6.2) because the layer thickness is only 0.03 mm. In this situation, the effect of volumetric error and surface roughness on the determined OBO is insignificant and what affect the determined OBO significantly are the remaining three attributes (i.e. support volume, build time and build cost).

The abnormal results of the sensitivity analysis experiment above reflect an obvious limitation of the developed method: The attributes considered in the method are not conflicting and are not diverse enough, so that the attribute weights lose their due role. To this end, other important conflicting attributes should be considered to improve the method once suitable estimation models are available in the future.

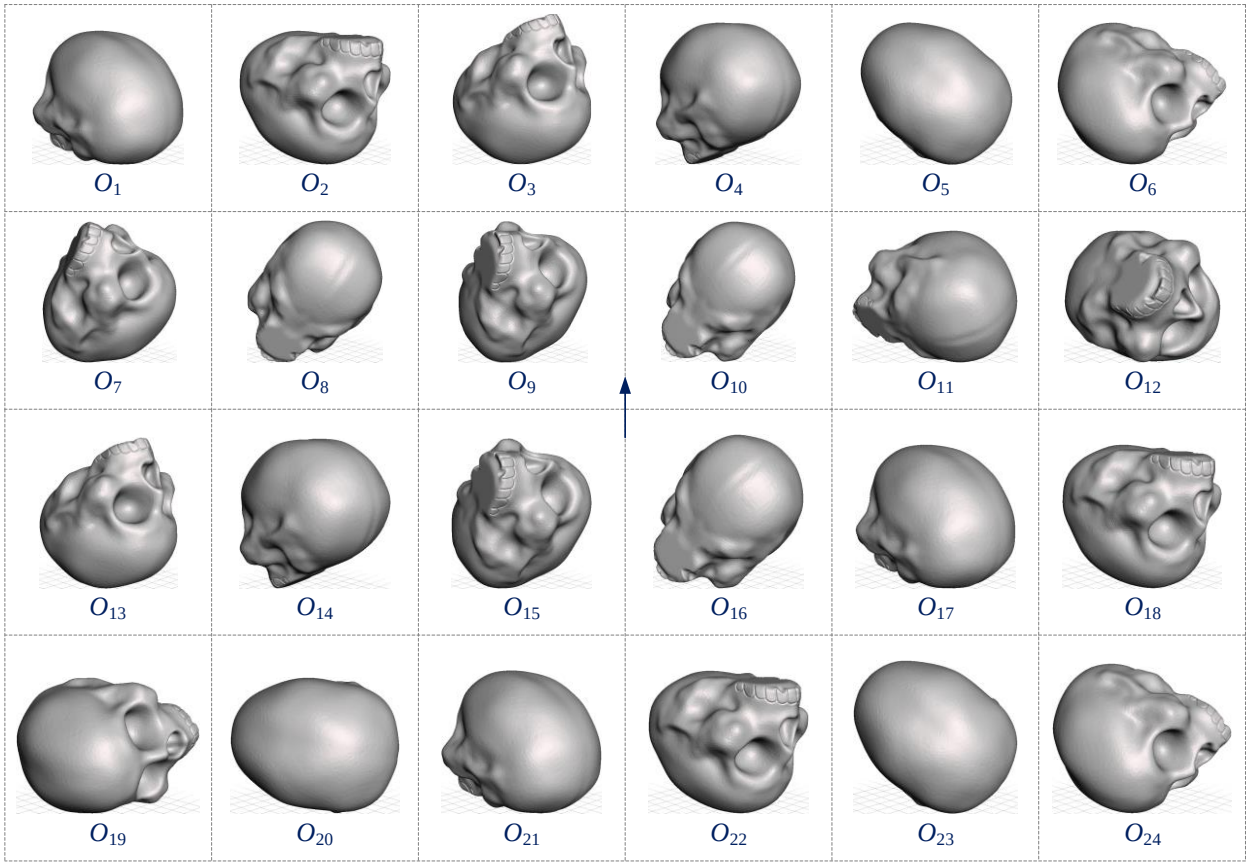
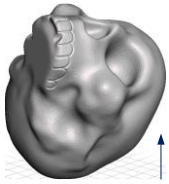


Figure 6.22 Schematic diagram of the generated ABOs of Part 10

$O_9 \succ O_3 \succ O_{15} \succ O_{13} \succ O_7 \succ O_1 \succ$
$O_2 \succ O_{22} \succ O_{18} \succ O_{10} \succ O_8 \succ O_{23} \succ$
$O_5 \succ O_{11} \succ O_{16} \succ O_{17} \succ O_{21} \succ O_{12} \succ$
$O_{20} \succ O_6 \succ O_{19} \succ O_{24} \succ O_{14} \succ O_4$

$O_9 = (-0.9746, +0.1419, +0.1731)$



Used LBPF machine:	Eosint M270
Used build material:	Ti6Al4V
Specified layer thickness:	0.03 mm
Specified recoating time:	20 s
Specified hatch spacing:	0.07 mm
Specified scanning velocity:	1,250 mm/s
Estimated support volume:	15,849.00 mm^3
Estimated volumetric error:	347.59 mm^3
Estimated surface roughness:	10.82 μm
Estimated build time:	79.26 h
Estimated build cost:	4,219.07 £

Figure 6.23 Build orientation determination results of Part 10

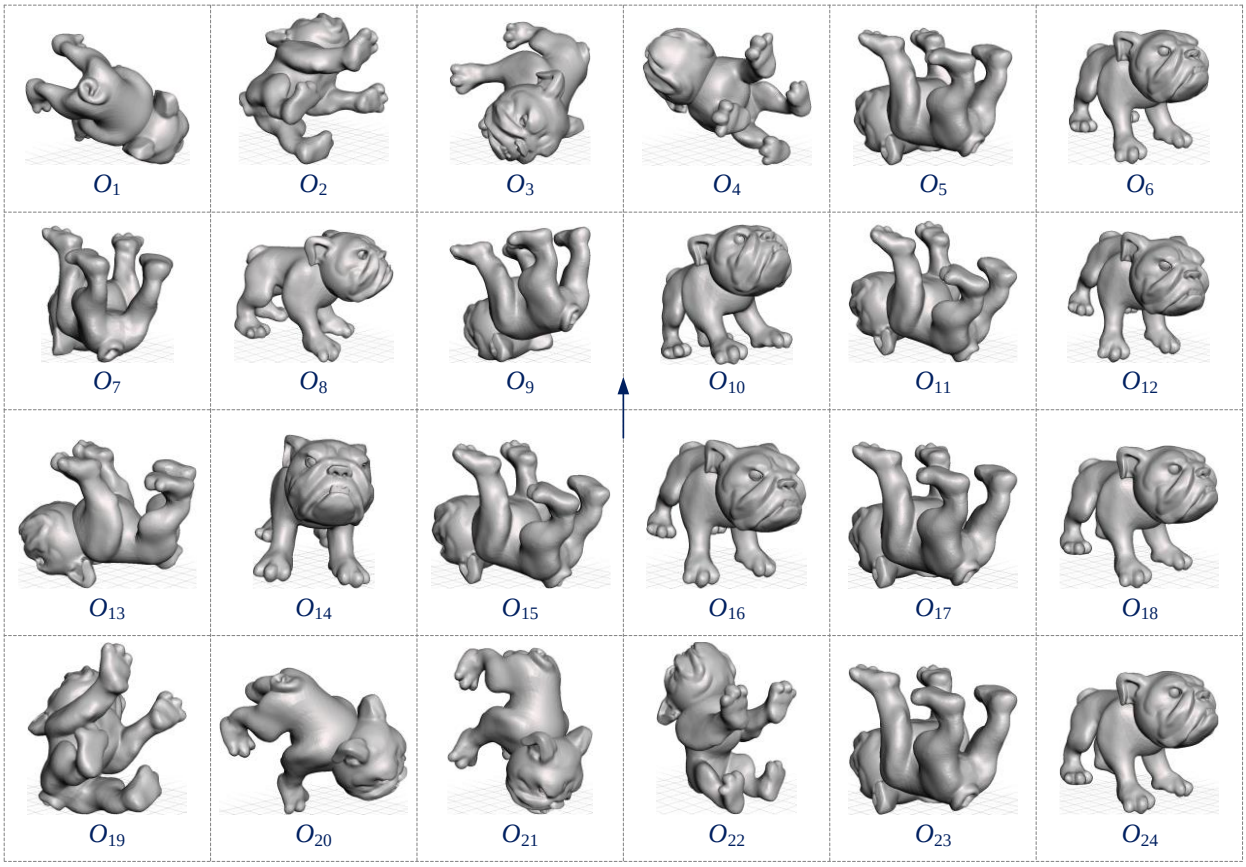


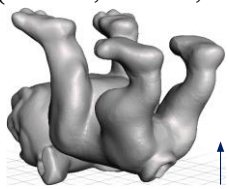
Figure 6.24 Schematic diagram of the generated ABOs of Part 11

6.3 Performance evaluation

In general, the main indicators used to evaluate the performance of a build orientation determination method include effectiveness, efficiency and advantages. In this section, the effectiveness and efficiency of the developed method are respectively assessed by theoretical analysis and experimental analysis. Then the advantages of the method over the existing methods are demonstrated via qualitative and quantitative comparisons.

$O_{17} \succ O_5 \succ O_{23} \succ O_6 \succ O_{24} \succ O_{21} \succ$
$O_{18} \succ O_8 \succ O_9 \succ O_{22} \succ O_{14} \succ O_7 \succ$
$O_{10} \succ O_{12} \succ O_{11} \succ O_{16} \succ O_{19} \succ O_{15} \succ$
$O_{13} \succ O_{20} \succ O_4 \succ O_3 \succ O_2 \succ O_1$

O_{17} = (+0.0884, +0.1803, +0.9796)



Used LBPF machine:	Eosint M270
Used build material:	Ti6Al4V
Specified layer thickness:	0.03 mm
Specified recoating time:	20 s
Specified hatch spacing:	0.07 mm
Specified scanning velocity:	1,250 mm/s
Estimated support volume:	10,913.80 mm^3
Estimated volumetric error:	164.34 mm^3
Estimated surface roughness:	10.70 µm
Estimated build time:	23.22 h
Estimated build cost:	1,154.69 £

Figure 6.25 Build orientation determination results of Part 11

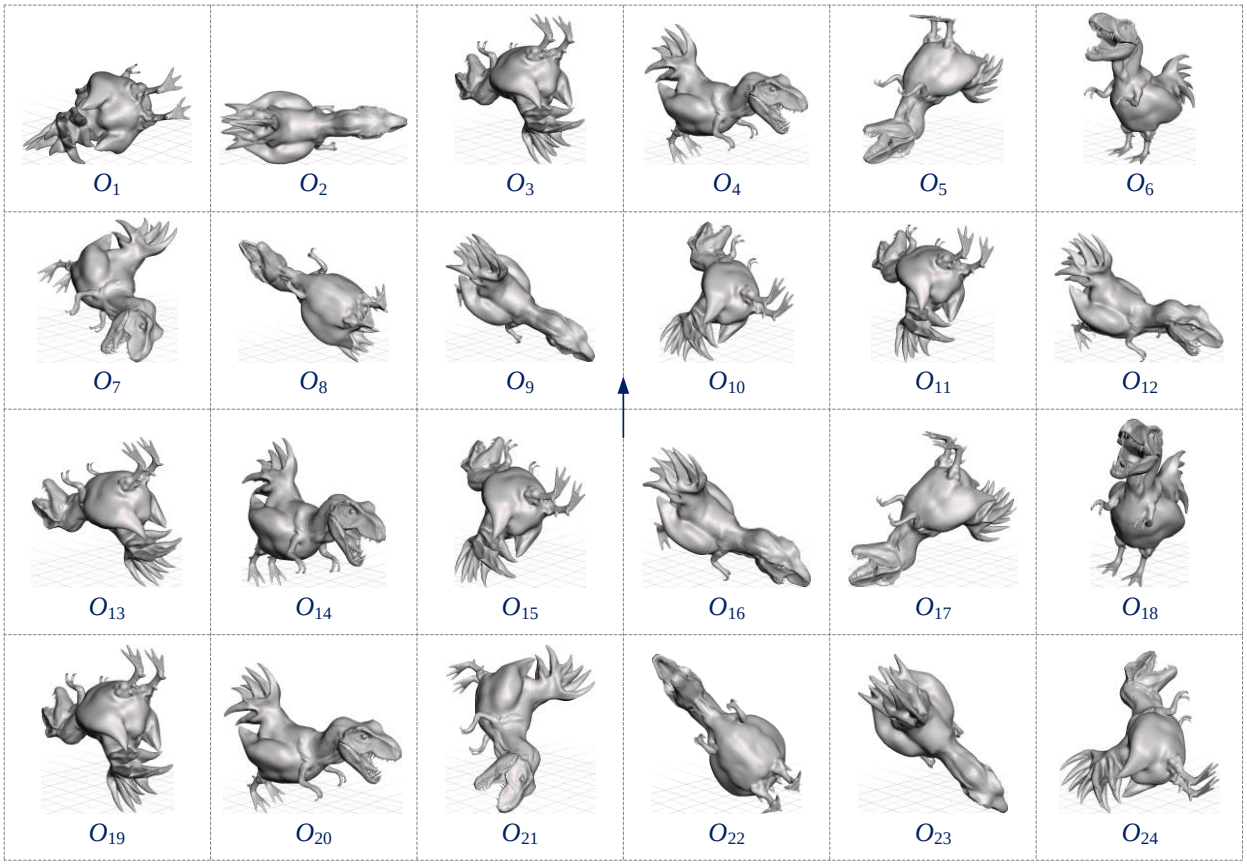


Figure 6.26 Schematic diagram of the generated ABOs of Part 12

6.3.1 Evaluation of the effectiveness

The main components of the developed automatic determination method in this chapter are the proposed facet clustering method in Chapter 3 and the proposed MADM method in Chapter 5. An evaluation of the effectiveness of the developed method is carried out via respectively assessing the effectiveness of the two proposed methods.

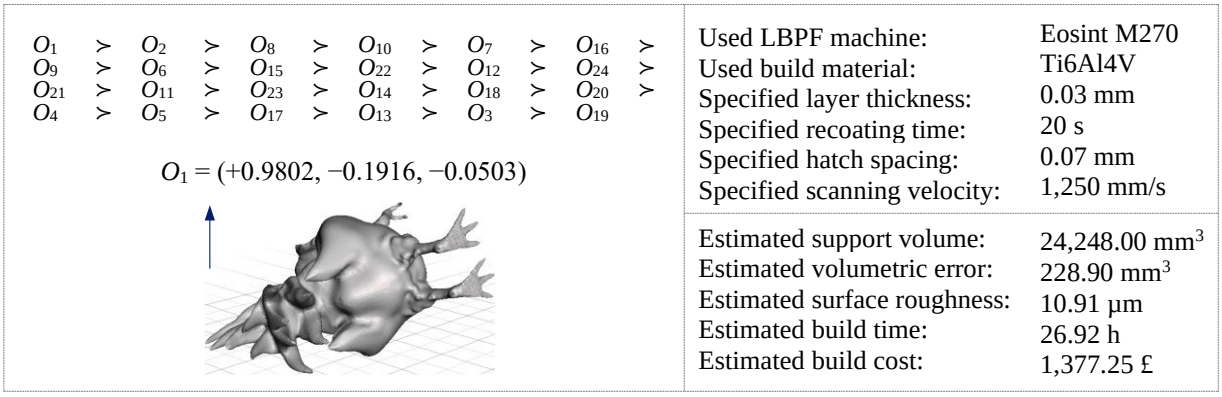


Figure 6.27 Build orientation determination results of Part 12

In general, whether a build orientation determination method is effective depends on whether the build orientation determined by it can meet the pre-set objective. The objective of the developed method is to concurrently optimise the total support volume, total volumetric error, average surface roughness, total build time and total build cost. It is found that the proposed facet clustering method can theoretically benefit the simultaneous optimisation of the total support volume, total build time and total build cost of an LPBF part. The reason is explained below.

The facet clustering rule (i.e. Rule 3.1), cluster refinement rule (i.e. Rule 3.3) and ABO generation rule (i.e. Rule 3.2) used in the proposed facet clustering method tend to generate the ABOs that allow as many facets as possible (or approximately) on the build platform. These facets will usually not become overhangs. Even if they become overhangs, the facets will require very few supports, because the distance between them and the build platform is very short. Based on this, the rules used are helpful for minimising the total support volume of an LPBF part. They are thus beneficial for optimisation of the total build time and total build cost of the part according to the estimation models in Equation (5.6) and Equation (5.8).

The proposed MADM method applies fuzzy AOs to determine a build orientation which can simultaneously optimise the total support volume, total volumetric error, average surface roughness, total build time and total build cost from the ABOs generated by the proposed facet clustering method. From this principle, it is not difficult to conclude that the proposed MADM method is theoretically effective.

6.3.2 Evaluation of the efficiency

In theory, the time complexity of the proposed facet clustering method is $O(n\log n)$ (where n is the number of facets), which is lower than the time complexity of the facet clustering method of Zhang et al. (2019), i.e. $O(n^2)$.

The used estimation models of volumetric error, surface roughness, build time and build cost on m ABOs and n facets have $O(mn)$ run-time. The constructed fuzzy AOs on m ABOs and n' attributes have $O(mn'!n')$ run-time. The algorithm for ranking m ABOs has $O(m^2)$ run-time. Consequently, the time complexity of the proposed MADM method is $O(mn+mn'!n'+m^2)$. This complexity looks a bit scary, but actually it is not high for the method, since the value of n' is 5 and the maximum value of m is 24 in the method.

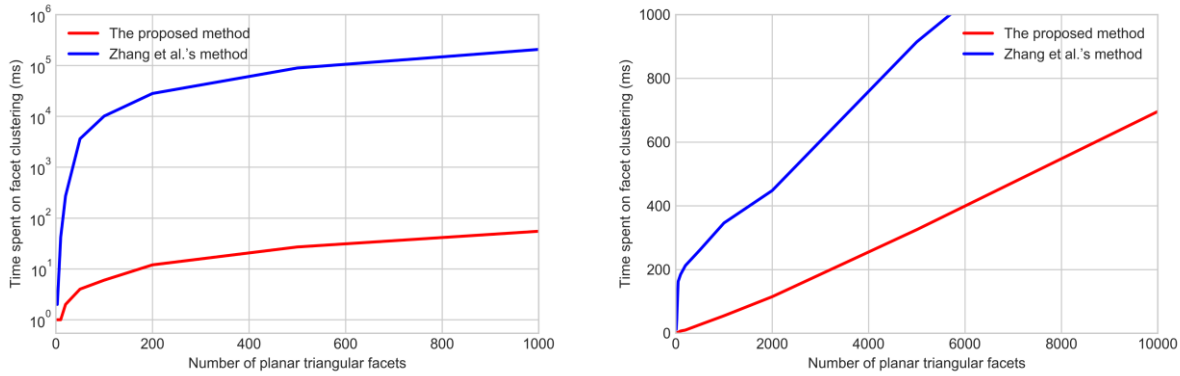
From the theoretical analysis above, it is difficult to imagine how efficient the proposed facet clustering method and MADM method are. To this end, four efficiency evaluation experiments were carried out on a machine with Intel(R) Core(TM) i9-7900X CPU, 8 GB RAM and NVIDIA GeForce GTX 1080 Ti GPU.

Experiment 6.1. The purpose of this experiment is to compare the facet clustering time taken by the facet clustering method of Zhang et al. (2019) and the proposed facet clustering method with respect to the number of facets. The method of Zhang et al. (2019) is the only ABO generation method based on facet clustering found in the literature. This method naturally becomes a comparison object of the proposed method. It was implemented and the availability of the source code is stated in Appendix E. Since the difference between the two methods is in facet clustering and both methods adopt the same rule to generated ABOs from facet clusters, only the time required in producing facet clusters is considered in the experiment for the sake of simplicity. Suppose the minimum Davies Bouldin value is always obtained from k = 2 to k = 40 for Zhang et al.'s method and a refinement is always needed for the proposed method. The time spent on facet clustering of the two methods from n = 1 to n = 1,000 is shown in Figure 6.28(a). It is obvious that the efficiency of the proposed method is higher than that of Zhang et al.'s method. The main reasons are as follows:

- The time complexity of the proposed method ($O(n\log n)$) is lower than that of Zhang et al.'s method ($O(n^2)$);
- The computation of Davies Bouldin values to find a proper value of k in Zhang et al.'s method is more time consuming than the computation on refinement of the produced facet clusters in the proposed method.
- The computation of the angles between normal vectors in Zhang et al.'s method also takes a lot of time. It is transformed to the more efficient computation of Euclidean distances in the proposed method.

Since the first two points cannot be changed, it is further assumed that the calculation of angles is converted to the calculation of Euclidean distances in Zhang et al.'s method. The compar-

ison result under such assumption is shown in Figure 6.28(b). It is found that the proposed method still outperforms Zhang et al.'s method. To further test the efficiency of the proposed method, *n* was assigned from 1 to 1,000,000 and the facet clustering time is depicted in Figure 6.29. As can been seen from the figure, the proposed method can provide acceptable efficiency when the number of facets is within 1,000,000.



(a) Angle metric is used in Zhang et al.'s method

(b) Euclidean distance metric is used in Zhang et al.'s method

Figure 6.28 Comparisons of the efficiency of the two facet clustering methods

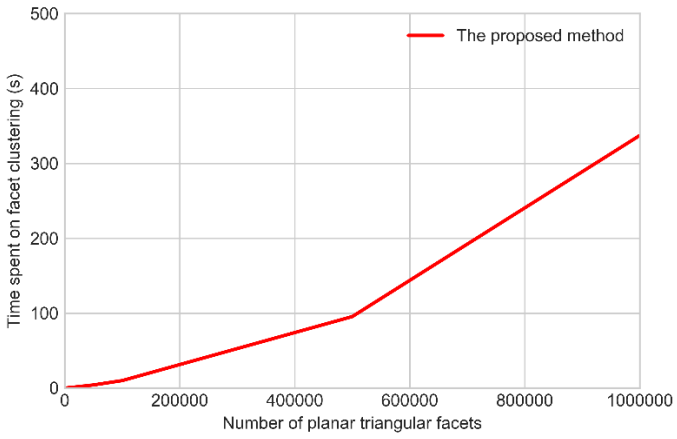


Figure 6.29 The efficiency of the proposed facet clustering method

Experiment 6.2. The purpose of this experiment is to compare the facet clustering time taken by the facet clustering method of Zhang et al. (2019) and the proposed facet clustering method with respect to specific STL models. In the experiment, the STL models of Part 1 to Part 12 in Figure 6.2 were respectively taken as the input of Zhang et al.'s method and the proposed method. Each of these methods was executed ten times for each STL model. The value of *k* in Zhang et al.'s method was assigned 6 for Part 1 to Part 8 and assigned 12 for Part 9 to Part 12 to make the output conditions of the two methods the same. The result of the experiment is the average

time spent on facet clustering of the two methods, which is listed in Table 6.8 and depicted in Figure 6.30. It can be seen from Table 6.8 and Figure 6.30 that the proposed method spends significantly less time on facet clustering than Zhang et al.'s method spends for the STL models of every part. This indicates that the proposed method is more efficient than Zhang et al.'s method in facet clustering.

Table 6.8 The facet clustering time taken by the two facet clustering methods

The input STL model (.stl)	**Part 1**	**Part 2**	**Part 3**	**Part 4**	**Part 5**	**Part 6**
Number of facets in the model	284	828	1,592	2,022	2,426	3,776
Time of Zhang et al.'s method (s)	0.3966	1.6378	3.2433	4.8869	6.4640	8.3093
Time of the proposed method (s)	0.0156	0.0625	0.1106	0.1457	0.1597	0.2895
The input STL model (.stl)	**Part 7**	**Part 8**	**Part 9**	**Part 10**	**Part 11**	**Part 12**
Number of facets in the model	6,324	6,598	13,240	67,722	88,724	108,888
Time of Zhang et al.'s method (s)	18.2422	24.7010	105.3338	425.3723	529.9664	575.2050
Time of the proposed method (s)	0.4538	0.4745	0.9607	5.9256	9.6683	10.6240

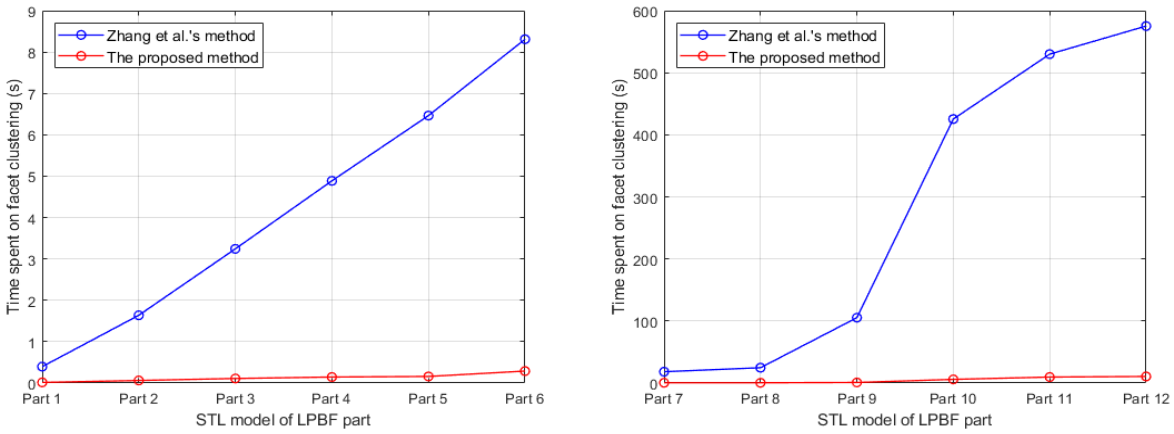


Figure 6.30 The facet clustering time of the two facet clustering methods

Experiment 6.3. The purpose of this experiment is to test the time spent on determination of the build orientations of specific LPBF parts for the developed automatic determination method. In the experiment, the STL models of the twelve parts in Figure 6.2 were taken as input of the developed method. For each STL model, the method was executed ten times. The results of the experiment are the average time spent on ABO generation, attribute value estimation and ranking generation for the method, which are listed in Table 6.9. It is worth noting that the time of ABO generation is the same as the time of facet clustering for the developed method, since the ABOs are generated in the process of facet clustering in the method. Further, the time of attribute value estimation does not include the time spent on estimation of support volume, because this estimation was completed by Autodesk Meshmixer and it is very difficult to count the exact time spent

on the estimation of the software tool. The average time of the entire build orientation determination process of the developed method is the sum of the average time of ABO generation, attribute value estimation and ranking generation, which is also listed in Table 6.9. As can be concluded from the table, the developed method can generate the OBO of a regular surface model within several seconds. It is also efficient for a freeform surface model.

Table 6.9 The actual build orientation determination time of the developed method

The input STL model (.stl)	**Part 1**	**Part 2**	**Part 3**	**Part 4**	**Part 5**	**Part 6**
Number of facets in the model	284	828	1,592	2,022	2,426	3,776
Number of the generated ABOs	6	6	6	6	8	8
Time of ABO generation (s)	0.0156	0.0625	0.1106	0.1457	0.1597	0.2895
Time of attribute estimation (s)	0.0984	0.2681	0.5140	0.6457	1.0292	1.5887
Time of ranking generation (s)	0.0601	0.0605	0.0593	0.0599	0.0622	0.0623
Time of the entire process (s)	0.1741	0.3911	0.6839	0.8513	1.2511	1.9405
The input STL model (.stl)	**Part 7**	**Part 8**	**Part 9**	**Part 10**	**Part 11**	**Part 12**
Number of facets in the model	6,324	6,598	13,240	67,722	88,724	108,888
Number of the generated ABOs	8	6	24	24	24	24
Time of ABO generation (s)	0.4538	0.4745	0.9607	5.9256	9.6683	10.6240
Time of attribute estimation (s)	2.6461	2.0775	16.5983	84.8202	110.7432	135.5582
Time of ranking generation (s)	0.0621	0.0599	0.0799	0.0793	0.0792	0.0800
Time of the entire process (s)	3.1620	2.6119	17.6389	90.8251	120.4907	146.2622

Experiment 6.4. The purpose of this experiment is to compare the build orientation determination time taken by an existing determination method and the developed automatic determination method with respect to specific STL models. As reviewed in Section 2.5, there are a large number of build orientation determination methods in the literature. It is not realistic to implement each method and compare its execution time with that of the developed method on specific STL models. However, it is still possible to compare the build orientation determination time of an existing method and the developed method with respect to specific STL models, because attribute value estimation is an indispensable task for each method and most of the build orientation determination time is spent on this task (see Table 6.9). The time of attribute value estimation is determined by the number of iterations (ABOs) for an exhaustive computation method or an MOO method (a feature recognition method, a facet clustering method or an MADM method) and the number of attributes. It is found from experiments that the effect of the number of iterations (ABOs) on the time of attribute value estimation is roughly uniform. If it is assumed that the effect of the number of ABOs on the time of ABO generation and the time of ranking generation in Table 6.9 is also roughly uniform, the time of the entire process per ABO calculated from the data in the table can be used to estimate the time of the entire build orientation determination process of an existing method. For example, as the time of the entire process for the STL model

of Part 1 is 0.1741 s and the number of the generated ABOs for Part 1 is 6, the time of the entire process per ABO for the STL model of Part 1 is 0.1741/6 s. Suppose an existing method needs to perform computation on 20 iterations (ABOs). The time of the entire process of this method for the STL model of Part 1 is 20 × 0.1741/6 s = 0.5803 s. By this way, the time of the entire process of an existing method for the STL models of the twelve parts in Figure 6.2 when the number of iterations (ABOs) m = 20, 40, 60, 80, 100 is estimated. The estimated results are listed in Table 6.10. Based on these results, a comparison of the build orientation determination time of an existing method and the developed method with respect to the twelve STL models is depicted in Figure 6.31. It can be concluded from Table 6.10 and Figure 6.31 that an existing method could outperform the developed method when m is smaller than the number of the generated ABOs for the STL model of each part in Table 6.10. This is possible for a feature recognition method or a facet clustering method, but most exhaustive computation methods and MOO methods are difficult to generate optimal solutions in such few iterations. For example, a particle swarm algorithm or a bacterial foraging algorithm for optimising AM process variables generally needs 100 iterations to obtain the optimal solutions according to the study of Raju et al. (2019). From this point of view, the developed method has obvious advantage in efficiency over an exhaustive computation method or an MOO method.

Table 6.10 The estimated build orientation determination time of an existing method

The input STL model (.stl)	**Part 1**	**Part 2**	**Part 3**	**Part 4**	**Part 5**	**Part 6**
Number of facets in the model	284	828	1,592	2,022	2,426	3,776
Number of the generated ABOs	6	6	6	6	8	8
Estimated time when m = 20 (s)	0.5803	1.3037	2.2797	2.8377	3.1278	4.8512
Estimated time when m = 40 (s)	1.1607	2.6073	4.5593	5.6753	6.2555	9.7025
Estimated time when m = 60 (s)	1.7410	3.9110	6.8390	8.5130	9.3833	14.5537
Estimated time when m = 80 (s)	2.3213	5.2147	9.1187	11.3507	12.5110	19.4050
Estimated time when m = 100 (s)	2.9017	6.5183	11.3983	14.1883	15.6388	24.2562
The input STL model (.stl)	**Part 7**	**Part 8**	**Part 9**	**Part 10**	**Part 11**	**Part 12**
Number of facets in the model	6,324	6,598	13,240	67,722	88,724	108,888
Number of the generated ABOs	8	6	24	24	24	24
Estimated time when m = 20 (s)	7.9050	8.7063	14.6991	75.6876	100.4089	121.8852
Estimated time when m = 40 (s)	15.8100	17.4127	29.3982	151.3752	200.8178	243.7703
Estimated time when m = 60 (s)	23.7150	26.1190	44.0973	227.0628	301.2268	365.6555
Estimated time when m = 80 (s)	31.6200	34.8253	58.7963	302.7503	401.6357	487.5407
Estimated time when m = 100 (s)	39.5250	43.5317	73.4954	378.4379	502.0446	609.4258

6.3.3 Evaluation of the advantages

Section 2.5 divided the existing ABO generation methods into exhaustive computation methods, feature recognition methods and facet clustering method and described their main strengths

and limitations. Briefly, the existing exhaustive computation methods can be applied to both regular and freeform surface models, but they generally need a large amount of computation to obtain desired results. The existing feature recognition methods can greatly reduce the amount of computation, but they are not applicable for freeform surface models. The existing facet clustering method can address the limitation of the existing feature recognition methods and maintain the advantage of the existing exhaustive computation methods, but it suffers from the issues of having relatively low efficiency, producing unstable results and working normally under specific distribution.

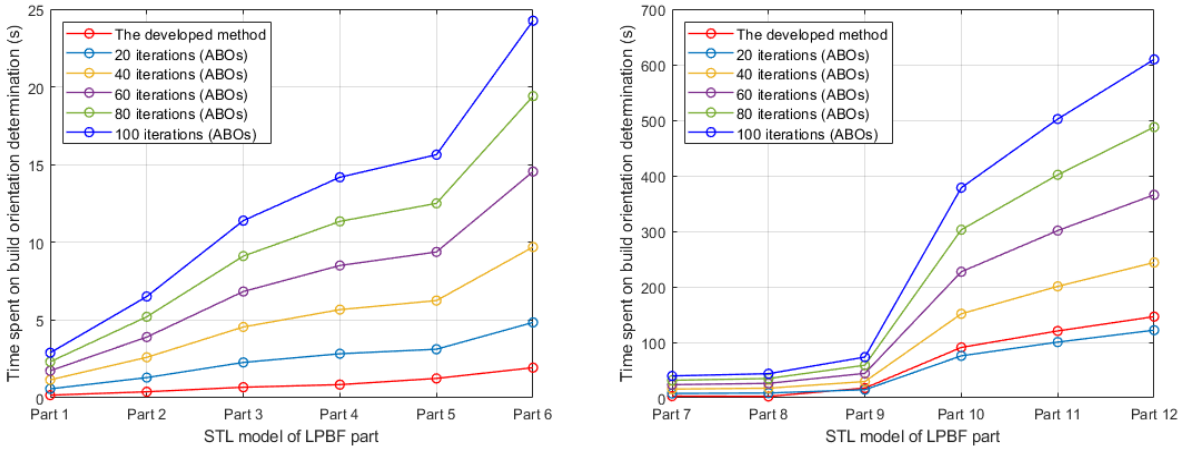


Figure 6.31 The efficiency of the developed method and an existing method

Compared with the existing exhaustive computation methods, the proposed ABO generation method in Chapter 3 does not need to spend time on the computation of a large number of meaningless orientations. This has been demonstrated in Experiment 6.4. Compared with the existing feature recognition methods, the proposed ABO generation method is applicable for both regular and freeform surface models. This has been explained in the specific process of the method in Section 3.1 and Section 3.2 and illustrated in Section 6.2. Compared with the existing facet clustering method (i.e. the method of Zhang et al. (2019)), the proposed ABO generation method has three advantages: providing higher efficiency; generating stable results; working normally under unknown probability distributions. The first advantage has been demonstrated in Experiment 6.1 and Experiment 6.2. The second and third advantages are respectively illustrated in the two experiments below.

Experiment 6.5. The purpose of this experiment is to illustrate that the proposed ABO generation method has advantage in generating stable results over the existing facet clustering meth-

od. In the experiment, the STL models depicted in Figure 6.32 were respectively input into the two methods. Each method was executed ten times for each STL model. The facet clustering results of the existing method and the proposed method are shown in Figure 6.33 and Figure 6.34, respectively. It should be noted that the facet clustering results of the proposed method obtained in each execution are identical for the same STL model. From Figure 6.33 and Figure 6.34, it can be seen that the facet clustering results of the existing method could vary in different executions and some of them are unreasonable results. This would result in different ABOs generated in different executions. Therefore, the results of the existing method are not stable. Unlike the existing method, the proposed method generates the same clusters and ABOs no matter how many times it is executed.

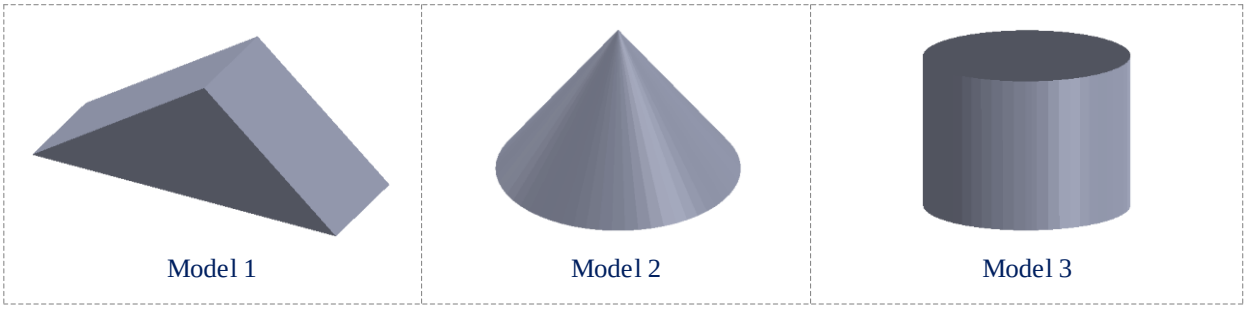


Figure 6.32 STL models for testing the stability of the two ABO generation methods

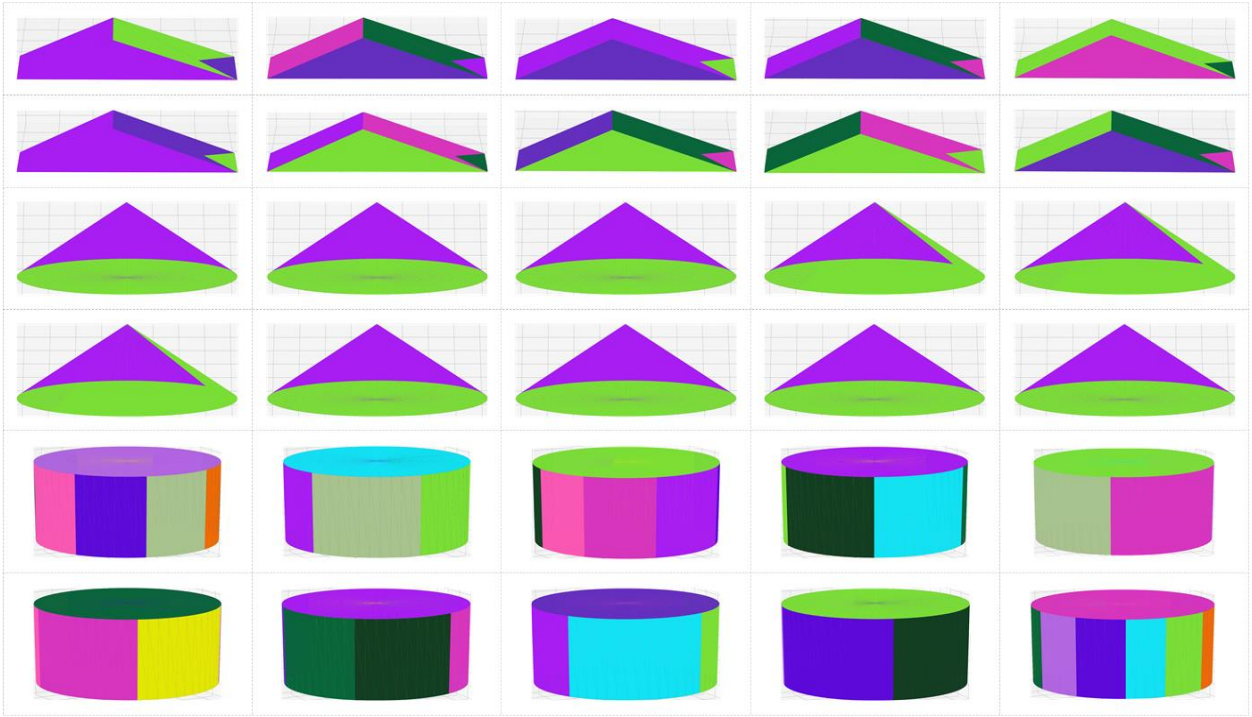


Figure 6.33 The clustering results of the existing facet clustering method

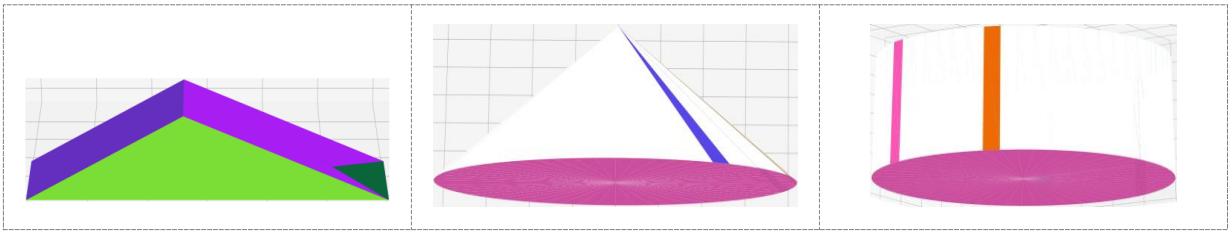


Figure 6.34 The clustering results of the proposed ABO generation method

Experiment 6.6. The purpose of this experiment is to illustrate that the proposed ABO generation method has advantage in working normally under unknown distributions over the existing facet clustering method. In the experiment, the STL models depicted in Figure 6.32 were respectively taken as the input of the two methods. The result of the experiment is shown in Figure 6.35. As can be seen from the figure, the existing method generate unreasonable results for Model 1 and Model 2 due to its random initialisation and distribution assumption. To be more specific, the Gaussian distribution assumption in the method could make the ends of all normal vectors in each cluster roughly spherical. The proposed method can always generate reasonable results for these models because there is no random initialisation and distribution assumption.

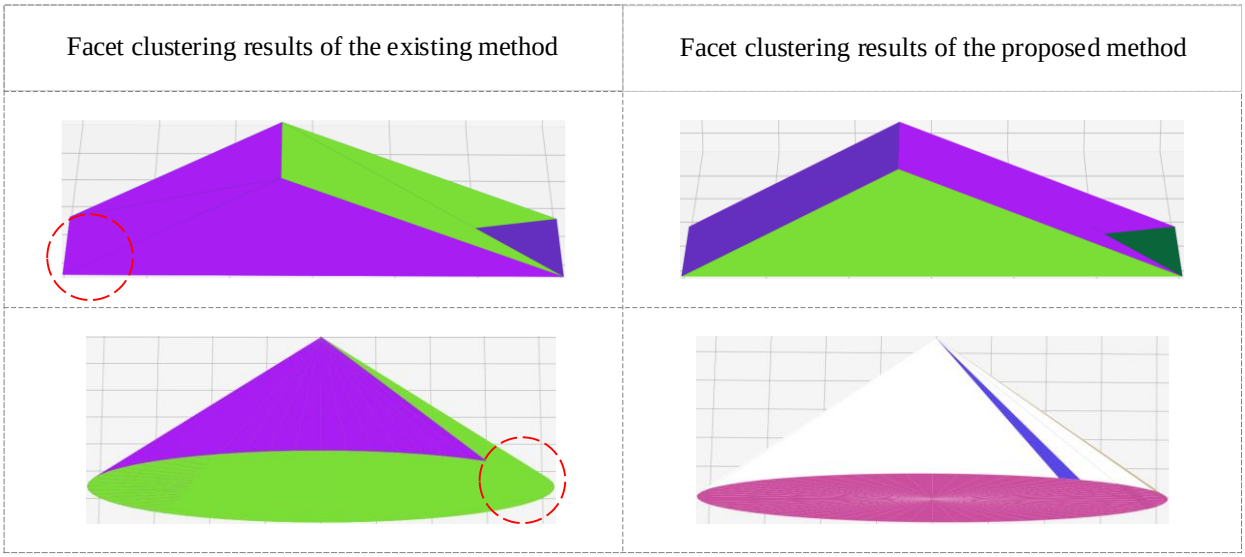


Figure 6.35 Comparison of the clustering results of the two ABO generation methods

Section 2.5 categorised the existing OBO selection methods into MOO methods and MADM methods and respectively described their main characteristics. In a nutshell, the computation cost of most MOO methods would be expensive when the number of ABOs or objectives increases. In addition, MOO algorithms could encounter difficulty in forming the Pareto front when dealing

with the selection cases with over three objectives and large number of optimal solutions on the Pareto front would bring difficulty to further selection. The MADM methods avoid these issues to some extent. In an MADM method, the computation time generally increases in a linear relationship with the number of ABOs or attributes. The number of the optimal solutions is usually one or many when the number of ABOs or attributes increases. However, the existing MADM methods have not captured the interrelationships of the considered attributes and have not lowered the negative influence of biased attribute values on the aggregation result.

Compared to the existing MOO methods, the proposed OBO selection method in Chapter 5 does not require expensive computation cost and will not generate those Pareto-optimal solutions that make selection difficult when the number of attributes is more than three. These can be respectively seen from the case studies in Section 6.2 and Experiment 6.4. Compared to the existing MADM methods, the proposed OBO selection method can capture the interrelationships of the considered attributes and reduce the negative effect of unreasonable estimated attribute values. These capabilities are illustrated in the experiment below.

Experiment 6.7. The purpose of this experiment is to illustrate that the proposed OBO selection method has advantages in capturing the interrelationships among the considered attributes and reducing the negative effect of unreasonable estimated attribute values. In this experiment, the method of Byun and Lee (2006), one of the most highly cited OBO selection methods, was used as an example of the OBO selection methods which assume all attributes are independent of each other and neglect the negative influence of biased attribute values on the aggregation result. To show the difference between the aggregation results of Byun and Lee's method and the proposed method intuitively, the STL models of Part 9 to Part 12 in Figure 6.2 were used as benchmarks. Assume the normalised build cost values of O_4 of Part 9, O_9 of Part 10, O_{17} of Part 11 and O_1 of Part 12 are biased attribute values. These values were constantly adjusted according to Table 6.11. It is easy to guess that such adjustments will respectively influence the ranking of O_4 of Part 9, O_9 of Part 10, O_{17} of Part 11 and O_1 of Part 12, each of which may probably be changed from the best ABO to the worst one. To confirm this conjecture, the score value of each of these four ABOs is respectively calculated using Byun and Lee's method and the proposed method. It should be mentioned that the score function in Equation (4.1) was used in both methods to facilitate the comparison. According to the calculated score values, the changes of the places of each of the four ABOs in the rankings generated by Byun and Lee's method and the proposed method are depicted in Figure 6.36. Please note that the scale values of each abscissa in Figure 6.36 are defined in Table 6.11.

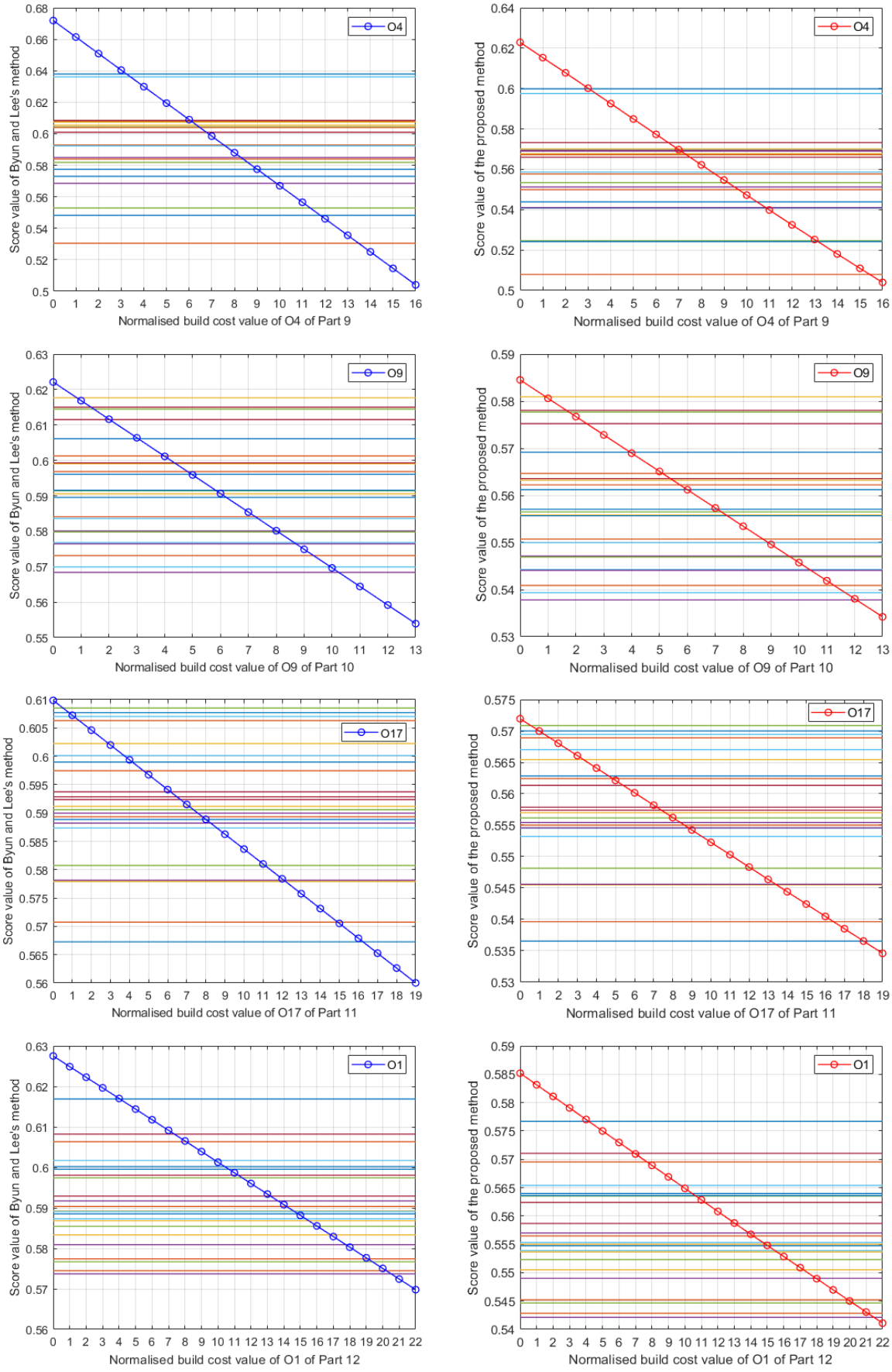


Figure 6.36 Comparison of the aggregation results of the two OBO selection methods

Table 6.11 The specific adjusted normalised build cost values

No.	Normalised build cost value of O_4 of Part 9		Normalised build cost value of O_9 of Part 10		Normalised build cost value of O_{17} of Part 11		Normalised build cost value of O_1 of Part 12	
	BLM	**TPM**	**BLM**	**TPM**	**BLM**	**TPM**	**BLM**	**TPM**
0	0.8360	<0.8360>	0.7941	<0.7941>	0.8039	<0.8039>	0.8390	<0.8390>
1	0.8160	<0.8160>	0.7841	<0.7841>	0.7989	<0.7989>	0.8340	<0.8340>
2	0.7960	<0.7960>	0.7741	<0.7741>	0.7939	<0.7939>	0.8290	<0.8290>
3	0.7760	<0.7760>	0.7641	<0.7641>	0.7889	<0.7889>	0.8240	<0.8240>
4	0.7560	<0.7560>	0.7541	<0.7541>	0.7839	<0.7839>	0.8190	<0.8190>
5	0.7360	<0.7360>	0.7441	<0.7441>	0.7789	<0.7789>	0.8140	<0.8140>
6	0.7160	<0.7160>	0.7341	<0.7341>	0.7739	<0.7739>	0.8090	<0.8090>
7	0.6960	<0.6960>	0.7241	<0.7241>	0.7689	<0.7689>	0.8040	<0.8040>
8	0.6760	<0.6760>	0.7141	<0.7141>	0.7639	<0.7639>	0.7990	<0.7990>
9	0.6560	<0.6560>	0.7041	<0.7041>	0.7589	<0.7589>	0.7940	<0.7940>
10	0.6360	<0.6360>	0.6941	<0.6941>	0.7539	<0.7539>	0.7890	<0.7890>
11	0.6160	<0.6160>	0.6841	<0.6841>	0.7489	<0.7489>	0.7840	<0.7840>
12	0.5960	<0.5960>	0.6741	<0.6741>	0.7439	<0.7439>	0.7790	<0.7790>
13	0.5760	<0.5760>	0.6641	<0.6641>	0.7389	<0.7389>	0.7740	<0.7740>
14	0.5560	<0.5560>	———	———	0.7339	<0.7339>	0.7690	<0.7690>
15	0.5360	<0.5360>	———	———	0.7289	<0.7289>	0.7640	<0.7640>
16	0.5160	<0.5160>	———	———	0.7239	<0.7239>	0.7590	<0.7590>
17	———	———	———	———	0.7189	<0.7189>	0.7540	<0.7540>
18	———	———	———	———	0.7139	<0.7139>	0.7490	<0.7490>
19	———	———	———	———	0.7089	<0.7089>	0.7440	<0.7440>
20	———	———	———	———	———	———	0.7390	<0.7390>
21	———	———	———	———	———	———	0.7340	<0.7340>
22	———	———	———	———	———	———	0.7290	<0.7290>

Notes: BLM: Byun and Lee's method; TPM: The proposed method

It can be seen from Figure 6.36 that the results of both methods are consistent with the conjecture. In addition, the ranking of each of the four ABOs generated by Byun and Lee's method drops to the last place faster than that generated by the proposed method under the same input: The ranking of O_4 of Part 9 generated by Byun and Lee's method drops to the last place at 14, while that generated by the proposed method drops to the last place at 16; The ranking of O_9 of Part 10 generated by Byun and Lee's method descends to the last place at 11, while that generated by the proposed method descends to the last place at 13; The ranking of O_{17} of Part 11 generated by Byun and Lee's method drops to the last place at 17, while that generated by the proposed method drops to the last place at 19; The ranking of O_1 of Part 12 generated by Byun and Lee's method descends to the last place at 21, while that generated by the proposed method descends to the last place at 22.

The reason for the results above is that the proposed method can adjust the weight values of the five attributes of each ABO dynamically as its normalised build cost value decreases, while Byun and Lee's method does not have such capability. To better explain this reason, Figure 6.37

visually displays the dynamic changes of the weight values of the five attributes of each ABO in the proposed method. Please note that the specific meanings of the scale values of each abscissa in this figure are also listed in Table 6.11. As can be seen from Figure 6.37, the weight value of the build cost of each ABO eventually drops and the weight values of the remaining four attributes of each ABO eventually rise. Due to such dynamic changes of the weights values, the negative influence of the adjustments of the normalised build cost value on the aggregation result is reduced. Compared to the proposed method, the weight values of the five attributes of each ABO are always 0.2443, 0.1402, 0.0909, 0.2623 and 0.2623 no matter how its normalised build cost value changes in Byun and Lee's method. Because of this, Byun and Lee's method does not have the capability to reduce the negative effect of the adjustments of the normalised build cost value.

To sum up, the difference of the aggregation results in Figure 6.36 and the dynamic changes of the weights values in Figure 6.37 are caused by the advantages of the proposed method in capturing the interrelationships among the considered attributes and reducing the negative influence of biased attribute values.

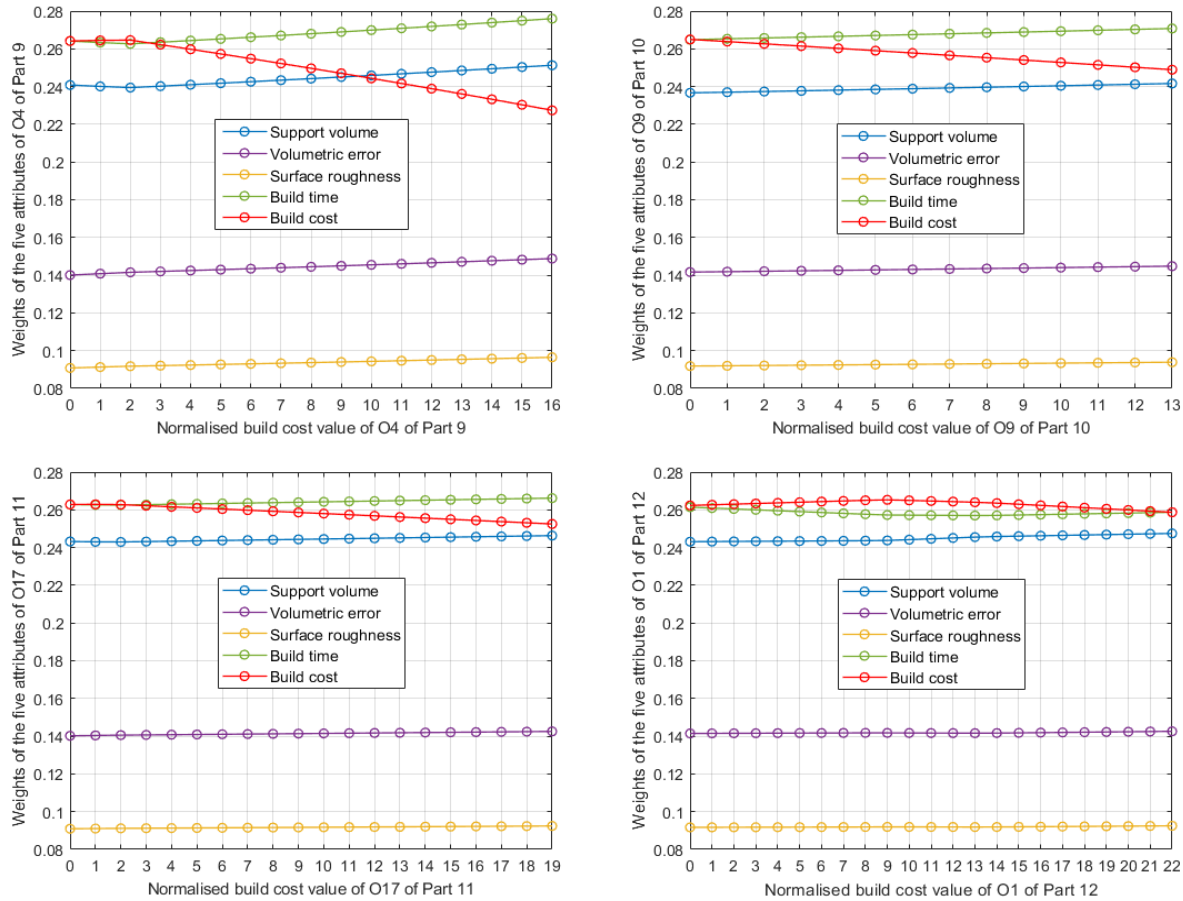


Figure 6.37 Dynamic changes of the weight values in the proposed method

6.4 Summary

This chapter has developed a method for automatic determination of part build orientation for LPBF AM. Twelve build orientation determination cases have been presented to demonstrate the developed automatic determination method. The effectiveness, efficiency and advantages of the developed method has been evaluated by theoretical analysis, experimental analysis and comparisons. The evaluation results suggest that the developed method is effective and efficient and can be applied to both regular and freeform surface models. In addition, the method has advantages in generating stable results, working normally under unknown probability distributions of facets, capturing the interrelationships among the considered attributes and lowering the negative effect of biased estimated attribute values over the existing build orientation determination methods.

7 Conclusion and future work

7.1 Conclusion

In response to the issue that the manually determined build orientation of an LPBF part could negatively affect the build time, build cost, quality and production stability of the part, this thesis studies the development of a method to automatically determine the build orientation of an LPBF part. The major contribution of the thesis lies in developing a methodology for automatic generation and evaluation of the ABOs of an LPBF part to determine the OBO of the part. The research work and specific contributions of the thesis are summarised as follows:

- A novel facet clustering method for automatically generating the ABOs of an LPBF part has been proposed. This method consists of a clustering of facets step and a generation of ABOs step. In the facet clustering step, the STL model of an LPBF part is imported. A facet clustering rule, the accelerated HDBSCAN* algorithm and the *k*-cluster lifetime partition criterion are applied to divide all facets of the imported STL model into a set of meaningful clusters. In the ABO generation step, an ABO generation rule is used to produce the ABOs of the part if a further refinement is not required for the obtained facet clusters. Otherwise, a cluster refinement rule is designed to refine the obtained facet clusters and the ABOs of the part are generated via the refined facet clusters and the ABO generation rule. From a theoretical perspective, the innovation of the proposed facet clustering method is reflected in the following three aspects: designing an effective rule to refine the generated facet clusters; finding a distance metric for the normal vectors of facets that can greatly accelerate the process of facet clustering; providing a feasible approach to identify clusters of facets from the output of a hierarchical clustering algorithm.
- An FAWPMM operator and an FAWPGMM operator for aggregating the values of attributes of ABOs have been constructed. The construction process of each AO consists of establishment of a general expression and derivation of specific expressions. In the establishment of a general expression, the formal definition of each AO is first presented. Its general expression is then established applying the operational laws of fuzzy numbers based on Archimedean t-norm and t-conorm. Based on the established general expression, the property and special cases of the AO are explored and discussed, respectively. In the derivation of specific expressions, the specific expressions of each AO are established through applying the AGs of ATT, ETT, HTT and FTT. The constructed AOs have good flexibility in the aggregation of fuzzy numbers and can be used to balance the opposite aggregation expectations. Most importantly,

they can capture the interrelationships of the aggregated fuzzy numbers and concurrently reduce the negative effect of biased fuzzy numbers on the aggregation result.

- A novel MADM method for automatically selecting the OBO of an LPBF part has been proposed. This method consists of an estimation of attribute values step, a normalisation of attribute values step and a selection of the OBO step. In the attribute value estimation step, the STL model and ABOs of an LPBF part, as well as the material, layer thickness, recoating time, hatch spacing and scanning velocity to build the part are first input into a set of estimation models. The total support volume, total volumetric error, average surface roughness, total build time and total build cost of the part in each ABO are then estimated. In the attribute value normalisation step, a ratio model is introduced to fuzzify the estimated attribute values. Based on this, a fuzzy decision matrix for selection of the OBO is established and normalised. In the OBO selection step, the relationships among support volume, volumetric error, surface roughness, build time and build cost are first identified. A scaling approach based on pairwise comparison is then used to determine the weights of these attributes. With the identified relationships and determined weights, the presented FAWPMM and FAWPGMM operators are leveraged to aggregate the normalised attribute values of each ABO. A ranking of all ABOs is generated based on the aggregation results. The OBO of the part is selected according to the generated ranking. From a theoretical perspective, the significance of the proposed MADM method is reflected in offering a fuzzy AOs-based framework for solving the MADM problems in manufacturing domain.
- A method for automatically determining the build orientation of an LPBF part has been developed and demonstrated. This method is composed of a module of generation of ABOs and a module of selection of the OBO, which are developed by implementing the proposed facet clustering method and the proposed MADM method, respectively. From a practical perspective, the significance of the developed method is reflected in providing a practical framework for determining the part build orientation in AM processes. Compared with the existing exhaustive computation methods, the developed method does not need to spend time on a large number of meaningless orientations. Compared to the existing feature recognition methods, the developed method is applicable for both regular and freeform surface models. Compared with the existing facet clustering method, the developed method can provide higher efficiency, produce more stable results and work normally with facet clusters of different probability distributions. Compared to the existing MOO methods, the developed method has a competitive efficiency and will not output the Pareto-optimal solutions that make selection difficult.

Compared to the existing MADM methods, the developed method can capture the interrelationships among attributes and lower the negative influence of extreme attribute values.

7.2 Future work

Several limitations of the developed automatic determination method of the build orientation of an LPBF part in the thesis revealed corresponding important research issues, which need to be addressed in further work. The research issues are outlined as follows:

- Consideration of the attributes related to LPBF part property. From the overall value chain, the property, accuracy and surface quality of an LPBF part are usually more important than the amount of supports, time and cost to build the part. The developed automatic determination method considers an attribute related to part accuracy (i.e. volumetric error), an attribute related to surface quality (i.e. surface roughness), support volume, build time and build cost. It neglects serval more important attributes that are related to part property, such as strength, elongation, hardness, residual stress, flexural modulus and fatigue performance. This directly causes the attribute weights in the proposed MADM method to lose its due role, since the effect of build orientation on the volumetric error and surface roughness of an LPBF part built by thin layers is quite slight and the estimated support volume, build time and build cost are positively correlated. Further, this is unacceptable for practical applications, because meeting the quality requirements is the most basic condition for practical applications and the attributes related to part property are critical part quality indicators. In the next research work, the developed method will be extended to consider part property attributes once suitable estimation models are available. New rules for facet clustering, ABO generation and cluster refinement which are beneficial to optimisation of the considered part property attributes would be developed. The five attributes considered in the method would be reduced to two or three attributes to add the part property attributes.
- Build orientation determination for multi-part production in LPBF AM. An important characteristic of the LPBF process is the support of multi-part production. The developed automatic determination method focuses on build orientation determination for production of one single part, it is not applicable for the situation where a group of parts in the same build need to be orientated simultaneously. It is of necessity and importance to improve the developed method to deal with this situation.
- Concurrent determination of all process variables to build an LPBF part. In LPBF AM, the influence of process variables (i.e. build orientation, supports, slices, laser scanning path and

process parameters) on part quality and production stability is comprehensive and contradictory. In the developed automatic determination method, the build orientation determination issue is addressed separately through fixing other process variables. This is sometimes insufficient, because the determination of build orientation could affect the determination of other process variables. An ideal approach is to simultaneously determine all process variables via taking into account certain part quality attributes. To this end, it would be desirable to extend the developed method to implement the concurrent determination of the process variables to build an LPBF part.

- Demonstration of effectiveness via actual part build experiments. Effectiveness is undoubtedly the most critical criterion for evaluating a build orientation determination method. The effectiveness of the developed automatic determination method is assessed through theoretical analysis. This would be insufficient, since there is a certain degree of uncertainty in the actual part build and a build orientation determination method that is effective in theory may not mean that it is also effective in practice. It is of necessity to carry out actual part build experiments to validate the developed method.
- Support of more 3D model data formats. The developed automatic determination method can only import the 3D models encoded by the STL format. Although this format is currently the most used and actual standard 3D model data format in LPBF AM, other alternative formats, such as the OBJ format, 3MF format and AMF format, also have certain applications because of their distinctive characteristics. It would be interesting to extend the developed method to make it applicable for more 3D model data formats.
- Realisation of industrial application. The ultimate goal of the research work in process planning for LPBF AM is to realise the application of novel methods in the industry. The developed automatic determination method is still at the stage of theoretical research and is still a long way from industrial application. It is believed that at least the five research issues above should be addressed and a software tool for process planning for LPBF AM should be developed and tested before applying the developed method to the industry.

Appendixes

Appendix A. Proof of Theorem 4.1

Proof:

Let

$$\varphi_{p(i)} = \left(w_{p(i)}\left(1+t(\alpha_{p(i)})\right)\right) \Big/ \sum_{j=1}^{n} w_j \left(1+t(\alpha_j)\right)$$

Then Equation (4.24) can be rewritten as

$$FAWPMM^{Q}(\alpha_1, \alpha_2, ..., \alpha_n) = \left(\frac{1}{n!}\bigoplus_{p\in \boldsymbol{P}_n}\bigotimes_{i=1}^{n}\left((n\varphi_{p(i)})\alpha_{p(i)}\right)^{q_i}\right)^{\frac{1}{\sum_{i=1}^{n} q_i}}$$

Using the operational law in Equation (4.15), it can be obtained that

$$(n\varphi_{p(i)})\alpha_{p(i)} = \left\langle g^{-1}\left((n\varphi_{p(i)})g(\mu_{p(i)})\right)\right\rangle$$

Using the operational law in Equation (4.16), it can be obtained that

$$\left((n\varphi_{p(i)})\alpha_{p(i)}\right)^{q_i} = \left\langle f^{-1}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(\mu_{p(i)})\right)\right)\right)\right\rangle$$

Using the operational law in Equation (4.14), it can be obtained that

$$\bigotimes_{i=1}^{n}\left((n\varphi_{p(i)})\alpha_{p(i)}\right)^{q_i} = \left\langle f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(\mu_{p(i)})\right)\right)\right)\right)\right\rangle$$

Using the operational law in Equation (4.13), it can be obtained that

$$\bigoplus_{p\in \boldsymbol{P}_n}\bigotimes_{i=1}^{n}\left((n\varphi_{p(i)})\alpha_{p(i)}\right)^{q_i} = \left\langle g^{-1}\left(\sum_{p\in \boldsymbol{P}_n} g\left(f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(\mu_{p(i)})\right)\right)\right)\right)\right)\right)\right\rangle$$

Using the operational law in Equation (4.15), it can be obtained that

$$\frac{1}{n!}\bigoplus_{p\in \boldsymbol{P}_n}\bigotimes_{i=1}^{n}\left((n\varphi_{p(i)})\alpha_{p(i)}\right)^{q_i} = \left\langle g^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} g\left(f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(\mu_{p(i)})\right)\right)\right)\right)\right)\right)\right\rangle$$

Using the operational law in Equation (4.16), it can be obtained that

$$\left(\frac{1}{n!}\bigoplus_{p\in \boldsymbol{P}_n}\bigotimes_{i=1}^{n}\left((n\varphi_{p(i)})\alpha_{p(i)}\right)^{q_i}\right)^{\frac{1}{\sum_{i=1}^{n} q_i}} =$$

$$\left\langle f^{-1}\left(\left(1\Big/\sum_{i=1}^{n} q_i\right) f\left(g^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} g\left(f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(\mu_{p(i)})\right)\right)\right)\right)\right)\right)\right)\right)\right\rangle$$

It is equivalent to the following equation

$$FAWPMM^{Q}(\alpha_1,\alpha_2,...,\alpha_n)=$$
$$\left\langle f^{-1}\left(\left(1\Big/\sum_{i=1}^{n}q_i\right)f\left(g^{-1}\left(\frac{1}{n!}\sum_{p\in\boldsymbol{P}_n}g\left(f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(\mu_{p(i)})\right)\right)\right)\right)\right)\right)\right)\right\rangle$$

It follows from the definition of a fuzzy number in Definition 4.2 that $0 \le \mu_{p(i)} \le 1$. Because $g(t)$ is a monotonically increasing function, it is can be obtained that

$$(n\varphi_{p(i)})g(0)\le(n\varphi_{p(i)})g(\mu_{p(i)})\le(n\varphi_{p(i)})g(1)$$

Because $g^{-1}(t)$ is a monotonically increasing function, it can be obtained that

$$g^{-1}\left((n\varphi_{p(i)})g(0)\right)\le g^{-1}\left((n\varphi_{p(i)})g(\mu_{p(i)})\right)\le g^{-1}\left((n\varphi_{p(i)})g(1)\right)$$

Because $f(t)$ is a monotonically decreasing function, it can be obtained that

$$\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(0)\right)\right)\right)\ge\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(\mu_{p(i)})\right)\right)\right)\ge\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(1)\right)\right)\right)$$

Because $f^{-1}(t)$ is a monotonically decreasing function, it can be obtained that

$$f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(0)\right)\right)\right)\right)\le f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(\mu_{p(i)})\right)\right)\right)\right)\le$$
$$f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(1)\right)\right)\right)\right)$$

Because $g(t)$ is a monotonically increasing function, it can be obtained that

$$\frac{1}{n!}\sum_{p\in\boldsymbol{P}_n}g\left(f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(0)\right)\right)\right)\right)\right)$$
$$\le\frac{1}{n!}\sum_{p\in\boldsymbol{P}_n}g\left(f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(\mu_{p(i)})\right)\right)\right)\right)\right)\le$$
$$\frac{1}{n!}\sum_{p\in\boldsymbol{P}_n}g\left(f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(1)\right)\right)\right)\right)\right)$$

Since

$$\sum_{p\in\boldsymbol{P}_n}\sum_{i=1}^{n}(n\varphi_{p(i)})=n\sum_{p\in\boldsymbol{P}_n}\sum_{i=1}^{n}\varphi_{p(i)}=n\frac{1}{n}=1$$

for the left and right sides of the inequality above, the inequality can be simplified as

$$g\left(f^{-1}\left(\left(\sum_{i=1}^{n}q_i\right)f(0)\right)\right)\le\frac{1}{n!}\sum_{p\in\boldsymbol{P}_n}g\left(f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(\mu_{p(i)})\right)\right)\right)\right)\right)\le$$
$$g\left(f^{-1}\left(\left(\sum_{i=1}^{n}q_i\right)f(1)\right)\right)$$

Because $g^{-1}(t)$ is a monotonically increasing function, it can be obtained that

$$f^{-1}\left(\left(\sum_{i=1}^{n} q_i\right) f(0)\right) \le g^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} g\left(f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(\mu_{p(i)})\right)\right)\right)\right)\right)\right) \le f^{-1}\left(\left(\sum_{i=1}^{n} q_i\right) f(1)\right)$$

Because $f(t)$ is a monotonically decreasing function, it can be obtained that

$$f(0) \ge \left(1\Big/\sum_{i=1}^{n} q_i\right) f\left(g^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} g\left(f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(\mu_{p(i)})\right)\right)\right)\right)\right)\right)\right) \ge f(1)$$

Because $f^{-1}(t)$ is a monotonically decreasing function, it can be obtained that

$$0 \le f^{-1}\left(\left(1\Big/\sum_{i=1}^{n} q_i\right) f\left(g^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} g\left(f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(\mu_{p(i)})\right)\right)\right)\right)\right)\right)\right)\right) \le 1$$

That is, $0 \le FAWPMM^{Q}(\alpha_1, \alpha_2, \ldots, \alpha_n) \le 1$. Therefore, $FAWPMM^{Q}(\alpha_1, \alpha_2, \ldots, \alpha_n)$ is still a fuzzy number. □

Appendix B. Proof of Theorem 4.2

Proof:

It follows from Theorem 4.1 that

$$FAWPMM^{Q}(\alpha_1, \alpha_2, \ldots, \alpha_n) =$$
$$\left\langle f^{-1}\left(\left(1\Big/\sum_{i=1}^{n} q_i\right) f\left(g^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} g\left(f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(\mu_{p(i)})\right)\right)\right)\right)\right)\right)\right)\right)\right\rangle$$

Since $\min(\mu_i) \le \mu_{p(i)} \le \max(\mu_i)$ and $g(t)$ is a monotonically increasing function, it can be obtained that

$$(n\varphi_{p(i)})g\left(\min(\mu_i)\right) \le (n\varphi_{p(i)})g(\mu_{p(i)}) \le (n\varphi_{p(i)})g\left(\max(\mu_i)\right)$$

Because $g^{-1}(t)$ is a monotonically increasing function, it can be obtained that

$$g^{-1}\left((n\varphi_{p(i)})g\left(\min(\mu_i)\right)\right) \le g^{-1}\left((n\varphi_{p(i)})g(\mu_{p(i)})\right) \le g^{-1}\left((n\varphi_{p(i)})g\left(\max(\mu_i)\right)\right)$$

Because $f(t)$ is a monotonically decreasing function, it can be obtained that

$$\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g\left(\min(\mu_i)\right)\right)\right)\right) \ge \sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(\mu_{p(i)})\right)\right)\right) \ge$$
$$\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g\left(\max(\mu_i)\right)\right)\right)\right)$$

Because $f^{-1}(t)$ is a monotonically decreasing function, it can be obtained that

$$f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g\left(\min(\mu_i)\right)\right)\right)\right)\right) \le f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(\mu_{p(i)})\right)\right)\right)\right) \le$$
$$f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g\left(\max(\mu_i)\right)\right)\right)\right)\right)$$

Because $g(t)$ is a monotonically increasing function, it can be obtained that

$$\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} g\left(f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)}) g\left(\min(\mu_i) \right) \right) \right) \right) \right) \right)$$
$$\leq \frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} g\left(f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)}) g(\mu_{p(i)}) \right) \right) \right) \right) \right) \leq$$
$$\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} g\left(f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)}) g\left(\max(\mu_i) \right) \right) \right) \right) \right) \right)$$

Since

$$\sum_{p\in \boldsymbol{P}_n}\sum_{i=1}^{n} (n\varphi_{p(i)}) = 1$$

for the left and right sides of the inequality above, the inequality can be simplified as

$$g\left(f^{-1}\left(\left(\sum_{i=1}^{n} q_i \right) f\left(\min(\mu_i) \right) \right) \right) \leq \frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} g\left(f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)}) g(\mu_{p(i)}) \right) \right) \right) \right) \right) \leq$$
$$g\left(f^{-1}\left(\left(\sum_{i=1}^{n} q_i \right) f\left(\max(\mu_i) \right) \right) \right)$$

Because $g^{-1}(t)$ is a monotonically increasing function, it can be obtained that

$$f^{-1}\left(\left(\sum_{i=1}^{n} q_i \right) f\left(\min(\mu_i) \right) \right) \leq g^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} g\left(f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)}) g(\mu_{p(i)}) \right) \right) \right) \right) \right) \right) \leq$$
$$f^{-1}\left(\left(\sum_{i=1}^{n} q_i \right) f\left(\max(\mu_i) \right) \right)$$

Because $f(t)$ is a monotonically decreasing function, it can be obtained that

$$f\left(\min(\mu_i) \right) \geq \left(1 \Big/ \sum_{i=1}^{n} q_i \right) f\left(g^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} g\left(f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)}) g(\mu_{p(i)}) \right) \right) \right) \right) \right) \right) \right) \geq$$
$$f\left(\max(\mu_i) \right)$$

Because $f^{-1}(t)$ is a monotonically decreasing function, it can be obtained that

$$\min(\mu_i) \leq f^{-1}\left(\left(1 \Big/ \sum_{i=1}^{n} q_i \right) f\left(g^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} g\left(f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)}) g(\mu_{p(i)}) \right) \right) \right) \right) \right) \right) \right) \right) \leq$$
$$\max(\mu_i)$$

Using Equation (4.1), the score values of $\alpha^{-} = \langle \min(\mu_i) \rangle$, $\alpha^{+} = \langle \max(\mu_i) \rangle$ and $FAWPMM^{Q}(\alpha_1, \alpha_2, \ldots, \alpha_n)$ are respectively calculated as follows:

$$S(\alpha^{-}) = 2\min(\mu_i) - 1$$

$$S(\alpha^{+}) = 2\max(\mu_i) - 1$$

$$S\left(FAWPMM^{Q}(\alpha_1, \alpha_2, ..., \alpha_n)\right) =$$

$$2f^{-1}\left(\left(1\Big/\sum_{i=1}^{n} q_i\right) f\left(g^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} g\left(f^{-1}\left(\sum_{i=1}^{n}\left(q_i f\left(g^{-1}\left((n\varphi_{p(i)})g(\mu_{p(i)})\right)\right)\right)\right)\right)\right)\right) - 1$$

Based on the inequality and equations above, it can be obtained that $S(\alpha^{-}) \le S(FAWPMM^{Q}(\alpha_1, \alpha_2, \ldots, \alpha_n)) \le S(\alpha^{+})$. From the comparison rule in Definition 4.4, it can be further obtained that $\alpha^{-} \le FAWPMM^{Q}(\alpha_1, \alpha_2, \ldots, \alpha_n) \le \alpha^{+}$. □

Appendix C. Proof of Theorem 4.3

Proof:

Using the operational law in Equation (4.16), it can be obtained that

$$\alpha_{p(i)}^{n\varphi_{p(i)}} = \left\langle f^{-1}\left((n\varphi_{p(i)}) f(\mu_{p(i)})\right)\right\rangle$$

Using the operational law in Equation (4.15), it can be obtained that

$$q_i\alpha_{p(i)}^{n\varphi_{p(i)}} = \left\langle g^{-1}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)}) f(\mu_{p(i)})\right)\right)\right)\right\rangle$$

Using the operational law in Equation (4.13), it can be obtained that

$$\bigoplus_{i=1}^{n}\left(q_i\alpha_{p(i)}^{n\varphi_{p(i)}}\right) = \left\langle g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)}) f(\mu_{p(i)})\right)\right)\right)\right)\right\rangle$$

Using the operational law in Equation (4.14), it can be obtained that

$$\bigotimes_{p\in \boldsymbol{P}_n}\bigoplus_{i=1}^{n}\left(q_i\alpha_{p(i)}^{n\varphi_{p(i)}}\right) = \left\langle f^{-1}\left(\sum_{p\in \boldsymbol{P}_n} f\left(g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)}) f(\mu_{p(i)})\right)\right)\right)\right)\right)\right)\right\rangle$$

Using the operational law in Equation (4.16), it can be obtained that

$$\left(\bigotimes_{p\in \boldsymbol{P}_n}\bigoplus_{i=1}^{n}\left(q_i\alpha_{p(i)}^{n\varphi_{p(i)}}\right)\right)^{\frac{1}{n!}} = \left\langle f^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} f\left(g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)}) f(\mu_{p(i)})\right)\right)\right)\right)\right)\right)\right\rangle$$

Using the operational law in Equation (4.15), it can be obtained that

$$\left(1\Big/\sum_{i=1}^{n} q_i\right)\left(\bigotimes_{p\in \boldsymbol{P}_n}\bigoplus_{i=1}^{n}\left(q_i\alpha_{p(i)}^{n\varphi_{p(i)}}\right)\right)^{\frac{1}{n!}} =$$

$$\left\langle g^{-1}\left(\left(1\Big/\sum_{i=1}^{n} q_i\right) g\left(f^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} f\left(g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)}) f(\mu_{p(i)})\right)\right)\right)\right)\right)\right)\right)\right)\right\rangle$$

It is equivalent to the following equation

$$FAWPGMM^{Q}(\alpha_1,\alpha_2,...,\alpha_n)=$$
$$\left\langle g^{-1}\left(\left(1\Big/\sum_{i=1}^{n}q_i\right)g\left(f^{-1}\left(\frac{1}{n!}\sum_{p\in\boldsymbol{P}_n}f\left(g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)})f(\mu_{p(i)})\right)\right)\right)\right)\right)\right)\right)\right\rangle$$

It follows from the definition of a fuzzy number in Definition 4.2 that $0 \le \mu_{p(i)} \le 1$. Because $f(t)$ is a monotonically decreasing function, it is can be obtained that

$$(n\varphi_{p(i)})f(0)\ge(n\varphi_{p(i)})f(\mu_{p(i)})\ge(n\varphi_{p(i)})f(1)$$

Because $f^{-1}(t)$ is a monotonically decreasing function, it can be obtained that

$$f^{-1}\left((n\varphi_{p(i)})f(0)\right)\le f^{-1}\left((n\varphi_{p(i)})f(\mu_{p(i)})\right)\le f^{-1}\left((n\varphi_{p(i)})f(1)\right)$$

Because $g(t)$ is a monotonically increasing function, it can be obtained that

$$\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)})f(0)\right)\right)\right)\le\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)})f(\mu_{p(i)})\right)\right)\right)\le\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)})f(1)\right)\right)\right)$$

Because $g^{-1}(t)$ is a monotonically increasing function, it can be obtained that

$$g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)})f(0)\right)\right)\right)\right)\le g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)})f(\mu_{p(i)})\right)\right)\right)\right)\le$$
$$g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)})f(1)\right)\right)\right)\right)$$

Because $f(t)$ is a monotonically decreasing function, it can be obtained that

$$\frac{1}{n!}\sum_{p\in\boldsymbol{P}_n}f\left(g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)})f(0)\right)\right)\right)\right)\right)$$
$$\ge\frac{1}{n!}\sum_{p\in\boldsymbol{P}_n}f\left(g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)})f(\mu_{p(i)})\right)\right)\right)\right)\right)\ge$$
$$\frac{1}{n!}\sum_{p\in\boldsymbol{P}_n}f\left(g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)})f(1)\right)\right)\right)\right)\right)$$

Since

$$\sum_{p\in\boldsymbol{P}_n}\sum_{i=1}^{n}(n\varphi_{p(i)})=n\sum_{p\in\boldsymbol{P}_n}\sum_{i=1}^{n}\varphi_{p(i)}=n\frac{1}{n}=1$$

for the left and right sides of the inequality above, the inequality can be simplified as

$$f\left(g^{-1}\left(\left(\sum_{i=1}^{n}q_i\right)g(0)\right)\right)\ge\frac{1}{n!}\sum_{p\in\boldsymbol{P}_n}f\left(g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)})f(\mu_{p(i)})\right)\right)\right)\right)\right)\ge$$
$$f\left(g^{-1}\left(\left(\sum_{i=1}^{n}q_i\right)g(1)\right)\right)$$

Because $f^{-1}(t)$ is a monotonically decreasing function, it can be obtained that

$$g^{-1}\left(\left(\sum_{i=1}^{n} q_i\right) g(0)\right) \le f^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} f\left(g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)}) f(\mu_{p(i)})\right)\right)\right)\right)\right)\right) \le g^{-1}\left(\left(\sum_{i=1}^{n} q_i\right) g(1)\right)$$

Because $g(t)$ is a monotonically increasing function, it can be obtained that

$$g(0) \le \left(1\Big/\sum_{i=1}^{n} q_i\right) g\left(f^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} f\left(g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)}) f(\mu_{p(i)})\right)\right)\right)\right)\right)\right)\right) \le g(1)$$

Because $g^{-1}(t)$ is a monotonically increasing function, it can be obtained that

$$0 \le g^{-1}\left(\left(1\Big/\sum_{i=1}^{n} q_i\right) g\left(f^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} f\left(g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)}) f(\mu_{p(i)})\right)\right)\right)\right)\right)\right)\right)\right) \le 1$$

That is, $0 \le FAWPGMM^{Q}(\alpha_1, \alpha_2, \ldots, \alpha_n) \le 1$. Therefore, $FAWPGMM^{Q}(\alpha_1, \alpha_2, \ldots, \alpha_n)$ is still a fuzzy number. □

Appendix D. Proof of Theorem 4.4

Proof:

It follows from Theorem 4.3 that

$$FAWPGMM^{Q}(\alpha_1, \alpha_2, \ldots, \alpha_n) =$$
$$\left\langle g^{-1}\left(\left(1\Big/\sum_{i=1}^{n} q_i\right) g\left(f^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} f\left(g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)}) f(\mu_{p(i)})\right)\right)\right)\right)\right)\right)\right)\right)\right\rangle$$

Since $\min(\mu_i) \le \mu_{p(i)} \le \max(\mu_i)$ and $f(t)$ is a monotonically decreasing function, it can be obtained that

$$(n\varphi_{p(i)}) f\left(\min(\mu_i)\right) \ge (n\varphi_{p(i)}) f(\mu_{p(i)}) \ge (n\varphi_{p(i)}) f\left(\max(\mu_i)\right)$$

Because $f^{-1}(t)$ is a monotonically decreasing function, it can be obtained that

$$f^{-1}\left((n\varphi_{p(i)}) f\left(\min(\mu_i)\right)\right) \le f^{-1}\left((n\varphi_{p(i)}) f(\mu_{p(i)})\right) \le f^{-1}\left((n\varphi_{p(i)}) f\left(\max(\mu_i)\right)\right)$$

Because $g(t)$ is a monotonically increasing function, it can be obtained that

$$\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)}) f\left(\min(\mu_i)\right)\right)\right)\right) \le \sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)}) f(\mu_{p(i)})\right)\right)\right) \le$$
$$\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)}) f\left(\max(\mu_i)\right)\right)\right)\right)$$

Because $g^{-1}(t)$ is a monotonically increasing function, it can be obtained that

$$g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)}) f\left(\min(\mu_i)\right)\right)\right)\right)\right) \le g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)}) f(\mu_{p(i)})\right)\right)\right)\right) \le$$
$$g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)}) f\left(\max(\mu_i)\right)\right)\right)\right)\right)$$

Because $f(t)$ is a monotonically decreasing function, it can be obtained that

$$\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} f\left(g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)})f\left(\min(\mu_i)\right)\right)\right)\right)\right)\right)$$
$$\geq \frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} f\left(g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)})f\left(\mu_{p(i)}\right)\right)\right)\right)\right)\right)\geq$$
$$\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} f\left(g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)})f\left(\max(\mu_i)\right)\right)\right)\right)\right)\right)$$

Since

$$\sum_{p\in \boldsymbol{P}_n}\sum_{i=1}^{n}(n\varphi_{p(i)}) = 1$$

for the left and right sides of the inequality above, the inequality can be simplified as

$$f\left(g^{-1}\left(\left(\sum_{i=1}^{n}q_i\right)g\left(\min(\mu_i)\right)\right)\right)\geq \frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} f\left(g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)})f\left(\mu_{p(i)}\right)\right)\right)\right)\right)\right)\geq$$
$$f\left(g^{-1}\left(\left(\sum_{i=1}^{n}q_i\right)g\left(\max(\mu_i)\right)\right)\right)$$

Because $f^{-1}(t)$ is a monotonically decreasing function, it can be obtained that

$$g^{-1}\left(\left(\sum_{i=1}^{n}q_i\right)g\left(\min(\mu_i)\right)\right)\leq f^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} f\left(g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)})f\left(\mu_{p(i)}\right)\right)\right)\right)\right)\right)\right)\leq$$
$$g^{-1}\left(\left(\sum_{i=1}^{n}q_i\right)g\left(\max(\mu_i)\right)\right)$$

Because $g(t)$ is a monotonically increasing function, it can be obtained that

$$g\left(\min(\mu_i)\right)\leq\left(1\Big/\sum_{i=1}^{n}q_i\right)g\left(f^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} f\left(g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)})f\left(\mu_{p(i)}\right)\right)\right)\right)\right)\right)\right)\right)\leq$$
$$g\left(\max(\mu_i)\right)$$

Because $g^{-1}(t)$ is a monotonically increasing function, it can be obtained that

$$\min(\mu_i)\leq g^{-1}\left(\left(1\Big/\sum_{i=1}^{n}q_i\right)g\left(f^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} f\left(g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)})f\left(\mu_{p(i)}\right)\right)\right)\right)\right)\right)\right)\right)\right)\leq$$
$$\max(\mu_i)$$

Using Equation (4.1), the score values of $\alpha^- = \langle\min(\mu_i)\rangle$, $\alpha^+ = \langle\max(\mu_i)\rangle$ and $FAWPGMM^Q(\alpha_1, \alpha_2, \ldots, \alpha_n)$ are respectively calculated as follows:

$$S(\alpha^-) = 2\min(\mu_i) - 1$$

$$S(\alpha^{+}) = 2\max(\mu_i) - 1$$

$$S\left(FAWPGMM^{Q}(\alpha_1, \alpha_2, ..., \alpha_n)\right) =$$

$$2g^{-1}\left(\left(1\Big/\sum_{i=1}^{n} q_i\right) g\left(f^{-1}\left(\frac{1}{n!}\sum_{p\in \boldsymbol{P}_n} f\left(g^{-1}\left(\sum_{i=1}^{n}\left(q_i g\left(f^{-1}\left((n\varphi_{p(i)}) f(\mu_{p(i)})\right)\right)\right)\right)\right)\right)\right) - 1$$

Based on the inequality and equations above, it can be obtained that $S(\alpha^{-}) \leq S(FAWPGMM^{Q}(\alpha_1, \alpha_2, \ldots, \alpha_n)) \leq S(\alpha^{+})$. From the comparison rule in Definition 4.4, it can be further obtained that $\alpha^{-} \leq FAWPGMM^{Q}(\alpha_1, \alpha_2, \ldots, \alpha_n) \leq \alpha^{+}$. □

Appendix E. Availability of code and data

The source code of the proposed methods and the compared methods, the STL files and the estimated attribute values used to support the research findings of this thesis have been deposited in the GitHub repository (https://github.com/YuchuChingQin/PartOrientationForLPBF).

References

Ahn, D., Kim, H. and Lee, S., 2007. Fabrication direction optimization to minimize post-machining in layered manufacturing. International Journal of Machine Tools and Manufacture, 47(3), 593−606.

Ahsan, A.N., Habib, M.A. and Khoda, B., 2015. Resource based process planning for additive manufacturing. Computer-Aided Design, 69, 112−125.

Ahsan, A.N. and Khoda, B., 2016. AM optimization framework for part and process attributes through geometric analysis. Additive Manufacturing, 11, 85−96.

Al-Ahmari, A., Abdulhameed, O. and Khan, A., 2018. An automatic and optimal selection of parts orientation in additive manufacturing. Rapid Prototyping Journal, 24(4), 698−708.

Alexander, P., Allen, S. and Dutta, D., 1998. Part orientation and build cost determination in layered manufacturing. Computer-Aided Design, 30(5), 343−356.

AMRG, 2020. About Additive Manufacturing. https://www.lboro.ac.uk/research/amrg/about/.

Ancau, M. and Caizar, C., 2010. The computation of Pareto-optimal set in multicriterial optimisation of rapid prototyping processes. Computers & Industrial Engineering, 58, 696−708.

Attaran, M., 2017. The rise of 3-D printing: The advantages of additive manufacturing over traditional manufacturing. Business Horizons, 60(5), 677−688.

Bartolomeu, F., Faria, S., Carvalho, O., Pinto, E., Alves, N., Silva, F.S. and Miranda, G., 2016. Predictive models for physical and mechanical properties of Ti6Al4V produced by Selective Laser Melting. Materials Science and Engineering: A, 663, 181−192.

Baumers, M., Tuck, C., Wildman, R., Ashcroft, I., Rosamond, E. and Hague, R., 2013. Transparency Built-in: Energy Consumption and Cost Estimation for Additive Manufacturing. Journal of Industrial Ecology, 17(3), 418−431.

Baumers, M., Dickens, P., Tuck, C. and Hague, R., 2016. The cost of additive manufacturing: Machine productivity, economies of scale and technology-push. Technological Forecasting and Social Change, 102, 193−201.

Blazdell, P., 2003. Solid free-forming of ceramics using a continuous jet printer. Journal of Materials Processing Technology, 137(1), 49−54.

Bourell, D.L., Leu, M.C. and Rosen, D.W., 2009. Roadmap for Additive Manufacturing: Identifying the Future of Freeform Processing. The University of Texas at Austin, Austin.

Brauers, W.K.M., Zavadskas, E.K., Peldschus, F. and Turskis, Z., 2008. Multi-objective decision-making for road design. Transport, 23(3), 183−193.

Brika, S.E., Zhao, Y.F., Brochu, M. and Mezzetta, J., 2017. Multi-Objective Build Orientation Optimisation for Powder Bed Fusion by Laser. Journal of Manufacturing Science and Engineering, 139(11), 111011.

Byun, H.S. and Lee, K.H., 2005. Determination of the optimal part orientation in layered manufacturing using a genetic algorithm. International Journal of Production Research, 43(13), 2709−2724.

Byun, H.S. and Lee, K.H., 2006. Determination of the optimal build direction for different rapid prototyping processes using multi-criterion decision making. Robotics and Computer-Integrated Manufacturing, 22(1), 69−80.

Calignano, F., 2018. Investigation of the accuracy and roughness in the laser powder bed fusion process. Virtual and Physical Prototyping, 13(2), 97−104.

Calvert, P., 2001. Inkjet printing for materials and devices. Chemistry of Materials, 13(10), 3299−3305.

Campello, R.J., Moulavi, D., Zimek, A. and Sander, J., 2015. Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection. ACM Transactions on Knowledge Discovery from Data, 10, 5.

Canellidis, V., Giannatsis, J. and Dedoussis, V., 2009. Genetic-algorithm-based multi objective optimization of the build orientation in stereolithography. The International Journal of Advanced Manufacturing Technology, 45(7−8), 714−730.

Chen, Y.H., Yang, Z.Y. and Ye, R.H., 2008. A fuzzy decision making approach to determine build orientation in automated layer-based machining. In: Proceedings of the 2008 IEEE International Conference on Automation and Logistics, IEEE, pp. 1−6.

Cheng, L. and To, A., 2019. Part-scale build orientation optimization for minimizing residual stress and support volume for metal additive manufacturing: Theory and experimental validation. Computer-Aided Design, 113, 1−23.

Cheng, W., Fuh, J.Y.H., Nee, A.Y.C., Wong, Y.S., Loh, H.T. and Miyazawa, T., 1995. Multi-objective optimisation of part-building orientation in stereolithography. Rapid Prototyping Journal, 1(4), 12−23.

Chowdhury, S., Mhapsekar, K. and Anand, S., 2018. Part Build Orientation Optimisation and Neural Network-Based Geometry Compensation for Additive Manufacturing Process. Journal of Manufacturing Science and Engineering, 140(3), 031009.

Davies, D.L. and Bouldin, D.W., 1979. A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1, 224−227.

De Gans, B.J., Duineveld, P.C. and Schubert, U.S., 2004. Inkjet Printing of Polymers: State of the Art and Future Developments. Advanced Materials, 16(3), 203−213.

Delfs, P., Tows, M. and Schmid, H.J., 2016. Optimized build orientation of additive manufactured parts for improved surface quality and build time. Additive Manufacturing, 12, 314−320.

Di Angelo, L., Di Stefano, P. and Guardiani, E., 2020. Search for the Optimal Build Direction in Additive Manufacturing Technologies: A Review. Journal of Manufacturing and Materials Processing, 4(3), 71.

Dowling, L., Kennedy, J., O'Shaughnessy, S. and Trimble, D., 2020. A review of critical repeatability and reproducibility issues in powder bed fusion. Materials & Design, 186, 108346.

Edwards, P. and Ramulu, M., 2014. Fatigue performance evaluation of selective laser melted Ti–6Al–4V. Materials Science and Engineering: A, 598, 327−337.

Elliott, A.M., Ivanova, O.S., Williams, C.B. and Campbell, T.A., 2013. Inkjet Printing of Quantum Dots in Photopolymer for Use in Additive Manufacturing of Nanocomposites. Advanced Engineering Materials, 15(10), 903−907.

Ezair, B., Massarwi, F. and Elber, G., 2015. Orientation analysis of 3D objects toward minimal support volume in 3D printing. Computers & Graphics, 51, 117−124.

Frank, D. and Fadel, G., 1995. Expert system-based selection of the preferred direction of build for rapid prototyping processes. Journal of Intelligent Manufacturing, 6(5), 339−345.

Fred, A.L. and Jain, A.K., 2005. Combining multiple clusterings using evidence accumulation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(6), 835−850.

Fritz, K. and Kim, I.Y., 2020. Simultaneous topology and build orientation optimization for minimization of additive manufacturing cost and time. International Journal for Numerical Methods in Engineering, 121(15), 3442−3481.

Galicia, J.A.G. and Benes, B., 2018. Improving printing orientation for Fused Deposition Modeling printers by analyzing connected components. Additive Manufacturing, 22: 720−728.

Ganeriwala, R.K., Strantza, M., King, W.E., Clausen, B., Phan, T.Q., Levine, L.E., Brown, D.W. and Hodge, N.E., 2019. Evaluation of a thermomechanical model for prediction of residual stress during laser powder bed fusion of Ti-6Al-4V. Additive Manufacturing, 27, 489−502.

Gao, W., Zhang, Y., Ramanujan, D., Ramani, K., Chen, Y., Williams, C.B., Wang, C.C., Shin, Y.C., Zhang, S. and Zavattieri, P.D., 2015. The status, challenges, and future of additive manufacturing in engineering. Computer-Aided Design, 69, 65−89.

Gibson, I., Rosen, D.W. and Stucker, B., 2015. Additive Manufacturing Technologies. Springer-Verlag New York, New York.

Golmohammadi, A.H. and Khodaygan, S., 2019. A framework for multi-objective optimisation of 3D part-build orientation with a desired angular resolution in additive manufacturing processes. Virtual and Physical Prototyping, 14(1), 19−36.

Grabisch, M., Marichal, J.L., Mesiar, R. and Pap, E., 2009. Aggregation Functions. Cambridge University Press, Cambridge.

Greco, S., Figueira, J. and Ehrgott, M., 2016. Multiple Criteria Decision Analysis: State of the Art Surveys. Springer-Verlag New York, New York.

Griffiths, V., Scanlan, J.P., Eres, M.H., Martinez-Sykora, A. and Chinchapatnam, P., 2019. Cost-driven build orientation and bin packing of parts in Selective Laser Melting (SLM). European Journal of Operational Research, 273(1), 334−352.

Hartigan, J.A. and Wong, M.A., 1979. Algorithm AS 136: A *k*-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28, 100−108.

Huang, R., Dai, N., Li, D., Cheng, X., Liu, H. and Sun, D., 2018. Parallel non-dominated sorting genetic algorithm-II for optimal part deposition orientation in additive manufacturing based on functional features. Journal of Mechanical Engineering Science, 232(19), 3384−3395.

Hur, J. and Lee, K., 1998. The development of a CAD environment to determine the preferred build-up direction for layered manufacturing. The International Journal of Advanced Manufacturing Technology, 14(4), 247−254.

Hur, S.M., Choi, K.H., Lee, S.H. and Chang, P.K., 2001. Determination of fabricating orientation and packing in SLS process. Journal of Materials Processing Technology, 112(2), 236−243.

ISO/ASTM 52900, 2015. Additive manufacturing—General principles—Terminology. International Organization for Standardization, Geneva.

ISO/ASTM 52904, 2019. Additive manufacturing—Process characteristics and performance—Practice for metal powder bed fusion process to meet critical applications. International Organization for Standardization, Geneva.

ISO/ASTM 52911-1, 2019. Additive manufacturing—Design—Part 1: Laser-based powder bed fusion of metals. International Organization for Standardization, Geneva.

ISO/ASTM 52911-2, 2019. Additive manufacturing—Design—Part 2: Laser-based powder bed fusion of polymers. International Organization for Standardization, Geneva.

ISO 4287, 1997. Geometrical Product Specifications (GPS)—Surface texture: Profile method—Terms, definitions and surface texture parameters. International Organization for Standardization, Geneva.

ISO 17296-2, 2015. Additive manufacturing—General principles—Part 2: Overview of process categories and feedstock. International Organization for Standardization, Geneva.

ISO 17296-3, 2014. Additive manufacturing—General principles—Part 3: Main characteristics and corresponding test methods. International Organization for Standardization, Geneva.

ISO 17296-4, 2014. Additive manufacturing—General principles—Part 4: Overview of data processing. International Organization for Standardization, Geneva.

Jaiswal, P., Patel, J. and Rai, R., 2018. Build orientation optimization for additive manufacturing of functionally graded material objects. The International Journal of Advanced Manufacturing Technology, 96(1−4), 223−235.

Jiang, J., Xu, X. and Stringer, J., 2018. Support Structures for Additive Manufacturing: A Review. Journal of Manufacturing and Materials Processing, 2(4), 64.

Jiang, J., Xu, X. and Stringer, J., 2019. Optimization of process planning for reducing material waste in extrusion based additive manufacturing. Robotics and Computer-Integrated Manufacturing, 59, 317−325.

Kahraman, C., 2008. Fuzzy Multi-Criteria Decision Making: Theory and Applications with Recent Developments. Springer Science & Business Media, New York.

Kim, H.C. and Lee, S.H., 2005. Reduction of post-processing for stereolithography systems by fabrication direction optimization. Computer-Aided Design, 37(7), 711−725.

Kim, D.B., Witherell, P., Lipman, R. and Feng, S.C., 2015. Streamlining the additive manufacturing digital spectrum: A systems approach. Additive Manufacturing, 5, 20−30.

King, W.E., Barth, H.D., Castillo, V.M., Gallegos, G.F., Gibbs, J.W., Hahn, D.E., Kamath, C. and Rubenchik, A.M., 2014. Observation of keyhole-mode laser melting in laser powder-bed fusion additive manufacturing. Journal of Materials Processing Technology, 214(12), 2915−2925.

King, W.E., Anderson, A.T., Ferencz, R.M., Hodge, N.E., Kamath, C., Khairallah, S.A. and Rubenchik, A.M., 2015. Laser powder bed fusion additive manufacturing of metals; physics, computational, and materials challenges. Applied Physics Reviews, 2(4), 041304.

Klement, E.P., Mesiar, R. and Pap, E., 2000. Triangular Norms. Springer, Dordrecht.

Klir, G.J. and Yuan, B., 1995. Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice Hall, Upper Saddle River.

Ko, S.H., Chung, J., Hotz, N., Nam, K.H. and Grigoropoulos, C.P., 2010. Metal nanoparticle direct inkjet printing for low-temperature 3D micro metal structure fabrication. Journal of Micromechanics and Microengineering, 20(12), 125010.

Kruth, J.P., 1991. Material Incress Manufacturing by Rapid Prototyping Techniques. CIRP Annals, 40(2), 603−614.

Kulkarni, P., Marsan, A. and Dutta, D., 2000. A review of process planning techniques in layered manufacturing. Rapid Prototyping Journal, 6(1), 18−35.

Lan, P.T., Chou, S.Y., Chen, L.L. and Gemmill, D., 1997. Determining fabrication orientations for rapid prototyping with stereolithography apparatus. Computer-Aided Design, 29(1), 53−62.

Li, Y. and Zhang, J., 2013. Multi-criteria GA-based Pareto optimization of building direction for rapid prototyping. The International Journal of Advanced Manufacturing Technology, 69(5−8), 1819−1831.

Li, C., Guo, Y., Fang, X. and Fang, F., 2018. A scalable predictive model and validation for residual stress and distortion in selective laser melting. CIRP Annals, 67(1), 249−252.

Li, P., Warner, D.H., Pegues, J.W., Roach, M.D., Shamsaei, N. and Phan, N., 2019. Towards predicting differences in fatigue performance of laser powder bed fused Ti-6Al-4V coupons from the same build. International Journal of Fatigue, 126, 284−296.

Liang, J.S., 2018. An ontology-oriented knowledge methodology for process planning in additive layer manufacturing. Robotics and Computer-Integrated Manufacturing, 53, 28−44.

Liu, P. and Wang, P., 2019. Multiple-Attribute Decision-Making Based on Archimedean Bonferroni Operators of *q*-Rung Orthopair Fuzzy Numbers. IEEE Transactions on Fuzzy Systems, 27(5), 834−848.

Luo, N. and Wang, Q., 2016. Fast slicing orientation determining and optimising algorithm for least volumetric error in rapid prototyping. The International Journal of Advanced Manufacturing Technology, 83(5−8), 1297−1313.

Mardani, A., Nilashi, M., Zavadskas, E.K., Awang, S.R., Zare, H. and Jamal, N.M., 2018. Decision Making Methods Based on Fuzzy Aggregation Operators: Three Decades Review from 1986 to 2017. International Journal of Information Technology & Decision Making, 17(2), 391−466.

Masood, S.H., Rattanawong, W. and Iovenitti, P., 2000. Part build orientations based on volumetric error in fused deposition modelling. The International Journal of Advanced Manufacturing Technology, 16(3), 162−168.

Masood, S.H., Rattanawong, W. and Iovenitti, P., 2003. A generic algorithm for a best part orientation system for complex parts in rapid prototyping. Journal of Materials Processing Technology, 139(1), 110−116.

Matos, M.A., Rocha, A.M.A. and Pereira, A.I., 2020. Improving additive manufacturing performance by build orientation optimization. The International Journal of Advanced Manufacturing Technology, 107(5−6), 1993−2005.

McClurkin, J.E. and Rosen, D.W., 1998. Computer-aided build style decision support for stereolithography. Rapid Prototyping Journal, 4(1), 4−13.

McInnes, L. and Healy, J., 2017. Accelerated Hierarchical Density Based Clustering. In: 2017 IEEE International Conference on Data Mining Workshops, IEEE, pp. 33−42.

Miranda, G., Faria, S., Bartolomeu, F., Pinto, E., Madeira, S., Mateus, A., Carreir, P., Alves, N., Silva, F.S. and Carvalho, O., 2016. Predictive models for physical and mechanical properties of 316L stainless steel produced by selective laser melting. Materials Science and Engineering: A, 657, 43−56.

Molitch-Hou, M., 2018. Overview of additive manufacturing process. In: Zhang, J. and Jung, Y.G. (Eds.) Additive Manufacturing: Materials, Processes, Quantifications and Applications, Butterworth-Heinemann, Oxford, pp. 1−38.

Mueller, B. and Kochan, D., 1999. Laminated object manufacturing for rapid tooling and patternmaking in foundry industry. Computers in Industry, 39(1), 47−53.

Muirhead, R.F., 1902. Some methods applicable to identities and inequalities of symmetric algebraic functions of *n* letters. Proceedings of the Edinburgh Mathematical Society, 21, 144−162.

Mukherjee, T., Zhang, W. and DebRoy, T., 2017. An improved prediction of residual stresses and distortion in additive manufacturing. Computational Materials Science, 126(1), 360−372.

Nesetril, J., Milkova, E. and Nesetrilova, H., 2001. Otakar Boruvka on minimum spanning tree problem translation of both the 1926 papers, comments, history. Discrete Mathematics, 233, 3−36.

Newman, S.T., Zhu, Z., Dhokia, V. and Shokrani, A., 2015. Process planning for additive and subtractive manufacturing technologies. CIRP Annals, 64(1), 467−470.

Nezhad, A.S., Barazandeh, F., Rahimi, A.R. and Vatani, M., 2010. Pareto-based optimization of part orientation in stereolithography. Journal of Engineering Manufacture, 224(10), 1591−1598.

Nguyen, C.H.P. and Choi, Y., 2020. Concurrent density distribution and build orientation optimization of additively manufactured functionally graded lattice structures. Computer-Aided Design, 127, 102884.

Padhye, N. and Deb, K., 2011. Multi-objective optimisation and multi-criteria decision making in SLS using evolutionary approaches. Rapid Prototyping Journal, 17(6), 458−478.

Pandey, P.M., Thrimurthulu, K. and Reddy, N.V., 2004. Optimal part deposition orientation in FDM by using a multicriteria genetic algorithm. International Journal of Production Research, 42(19), 4069−4089.

Paul, R., and Anand, S., 2011. Optimal part orientation in rapid manufacturing process for achieving geometric tolerances. Journal of Manufacturing Systems, 30(4), 214−222.

Paul, R. and Anand, S., 2015. Optimization of layered manufacturing process for reducing form errors with minimal support structures, Journal of Manufacturing Systems, 36, 231−243.

Pereira, S., Vaz, A.I.F. and Vicente, L.N., 2018. On the optimal object orientation in additive manufacturing. The International Journal of Advanced Manufacturing Technology, 98(5−8), 1685−1694.

Pham, D.T., Dimov, S.S. and Gault, R.S., 1999. Part orientation in stereolithography. The International Journal of Advanced Manufacturing Technology, 15(9), 674−682.

Phatak, A.M. and Pande, S.S., 2012. Optimum part orientation in rapid prototyping using genetic algorithm. Journal of Manufacturing Systems, 31(4), 395−402.

Qie, L., Jing, S., Lian, R., Chen, Y. and Liu, J., 2018. Quantitative suggestions for build orientation selection. The International Journal of Advanced Manufacturing Technology, 98(5−8), 1831−1845.

Raju, M., Gupta, M.K., Bhanot, N. and Sharma, V.S., 2019. A hybrid PSO−BFO evolutionary algorithm for optimisation of fused deposition modelling process parameters. Journal of Intelligent Manufacturing, 30(7), 2743−2758.

Ram, P., Lee, D., March, W. and Gray, A.G., 2009. Linear-time algorithms for pairwise statistical problems. In: Bengio, Y., Schuurmans, D., Lafferty, J.D., Williams, C.K.I. and Culotta, A. (Eds.) Advances in Neural Information Processing Systems 22, Curran Associates, Inc., New York, pp. 1527−1535.

Ransikarbum, K. and Kim, N., 2017. Multi-criteria selection problem of part orientation in 3D fused deposition modelling based on analytic hierarchy process model: A case study. In: Proceedings of the 2017 IEEE International Conference on Industrial Engineering and Engineering Management, IEEE, pp. 1455−1459.

Rattanawong, W., Masood, S.H. and Iovenitti, P., 2001. A volumetric approach to part-build orientations in rapid prototyping. Journal of Materials Processing Technology, 119(1−3), 348−353.

Roscoe, L.E., 1988. Stereolithography Interface Specification. 3D Systems, Inc.

Ruffo, M., Tuck, C. and Hague, R., 2006. Cost estimation for rapid manufacturing-laser sintering production for low to medium volumes. Journal of Engineering Manufacture, 220(9), 1417−1427.

Saaty, T.L., 1977. A scaling method for priorities in hierarchical structures. Journal of Mathematical Psychology, 15(3), 234−281.

Senvol LLC, 2020. Senvol Database: Industrial Additive Manufacturing Machines and Materials. http://senvol.com/database/.

Shen, H., Ye, X., Xu, G., Zhang, L., Qian, J. and Fu, J., 2020. 3D printing build orientation optimization for flexible support platform. Rapid Prototyping Journal, 26(1), 59−72.

Shipley, H., McDonnell, D., Culleton, M., Coull, R., Lupoi, R., O'Donnell, G. and Trimble, D., 2018. Optimisation of process parameters to address fundamental challenges during selective laser melting of Ti-6Al-4V: A review. International Journal of Machine Tools and Manufacture, 128, 1−20.

Sing, S.L. and Yeong, W.Y., 2020. Laser powder bed fusion for metal additive manufacturing: perspectives on recent developments. Virtual and Physical Prototyping, 15(3), 359−370.

Singhal, S.K., Pandey, A.P., Pandey, P.M. and Nagpal, A.K., 2005. Optimum part deposition orientation in stereolithography. Computer-Aided Design and Applications, 2, 319−328.

Singhal, S.K., Jain, P.K., Pandey, P.M. and Nagpal, A.K., 2009. Optimum part deposition orientation for multiple objectives in SL and SLS prototyping. International Journal of Production Research, 47(22), 6375−6396.

Snyder, J.C., Stimpson, C.K., Thole, K.A. and Mongillo, D.J., 2015. Build Direction Effects on Microchannel Tolerance and Surface Roughness. Journal of Mechanical Design, 137(11), 111411.

Strano, G., Hao, L., Everson, R.M. and Evans, K.E., 2011. Multi-objective optimization of selective laser sintering processes for surface quality and energy saving. Journal of Engineering Manufacture, 225(9), 1673−1682.

Strano, G., Hao, L., Everson, R.M. and Evans, K.E., 2013. Surface roughness analysis, modelling and prediction in selective laser melting. Journal of Materials Processing Technology, 213(4), 589−597.

Tang, Y. and Zhao, Y., 2016. A survey of the design methods for additive manufacturing to improve functional performance. Rapid Prototyping Journal, 22(3), 569−590.

Tang, M., Pistorius, P.C. and Beuth, J.L., 2017. Prediction of lack-of-fusion porosity for powder bed fusion. Additive Manufacturing, 14, 39−48.

Thompson, M.K., Moroni, G., Vaneker, T., Fadel, G., Campbell, R.I., Gibson, I., Bernard, A., Schulz, J., Graf, P., Ahuja, B. and Martina, F., 2016. Design for Additive Manufacturing: Trends, opportunities, considerations, and constraints. CIRP Annals, 65(2), 737−760.

Thrimurthulu, K.P., Pandey, P.M. and Reddy, N.V., 2004. Optimum part deposition orientation in fused deposition modelling. International Journal of Machine Tools and Manufacture, 44(6), 585−594.

Townsend, A., Senin, N., Blunt, L., Leach, R.K. and Taylor, J.S., 2016. Surface texture metrology for metal additive manufacturing: A review. Precision Engineering, 46, 34−47.

Ulu, E., Gecer Ulu, N., Hsiao, W. and Nelaturi, S., 2020. Manufacturability Oriented Model Correction and Build Direction Optimization for Additive Manufacturing. Journal of Mechanical Design, 142(6), 062001.

Vaneker, T., Bernard, A., Moroni, G., Gibson, I. and Zhang, Y., 2020. Design for additive manufacturing: Framework and methodology. CIRP Annals, 69(2), 578−599.

Wang, C. and Qian, X., 2020. Simultaneous optimization of build orientation and topology for additive manufacturing. Additive Manufacturing, 34, 101246.

Wang, W., Shao, H., Liu, X. and Yin, B., 2020. Printing Direction Optimization Through Slice Number and Support Minimization. IEEE Access, 8, 75646−75655.

Wauthle, R., Vrancken, B., Beynaerts, B., Jorissen, K., Schrooten, J., Kruth, J.P. and Van Humbeeck, J., 2015. Effects of build orientation and heat treatment on the microstructure and mechanical properties of selective laser melted Ti6Al4V lattice structures. Additive Manufacturing, 5, 77−84.

Weber, C., Pena, V., Micali, M., Yglesias, E., Rood, S., Scott, J.A. and Lal, B., 2013. The Role of the National Science Foundation in the Origin and Evolution of Additive Manufacturing in the United States. Science & Technology Policy Institute, Washington, DC.

West, A.P., Sambu, S.P. and Rosen, D.W., 2001. A process planning method for improving build performance in stereolithography. Computer-Aided Design, 33(1), 65−79.

Williams, C.B., Mistree, F. and Rosen, D.W., 2011. A Functional Classification Framework for the Conceptual Design of Additive Manufacturing Technologies. Journal of Mechanical Design, 133(12), 121002.

Williams, R.J., Davies, C.M. and Hooper, P.A., 2018. A pragmatic part scale model for residual stress and distortion prediction in powder bed fusion. Additive Manufacturing, 22, 416−425.

Wohlers, T. and Gornet, T., 2016. History of Additive Manufacturing. Wohlers Associates, Inc.

Xia, M., Xu, Z. and Zhu, B., 2012. Some issues on intuitionistic fuzzy aggregation operators based on Archimedean t-conorm and t-norm. Knowledge-Based Systems, 31, 78−88.

Xu, F., Wong, Y.S., Loh, H.T., Fuh, J.Y.H. and Miyazawa, T., 1997. Optimal orientation with variable slicing in stereolithography. Rapid Prototyping Journal, 3(3), 76−88.

Xu, F., Loh, H.T. and Wong, Y.S., 1999. Considerations and selection of optimal orientation for different rapid prototyping systems. Rapid Prototyping Journal, 5(2), 54−60.

Yager, R.R., 2001. The Power Average Operator. IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans, 31(6), 724−731.

Yap, C.Y., Chua, C.K., Dong, Z.L., Liu, Z.H., Zhang, D.Q., Loh, L.E. and Sing, S.L., 2015. Review of selective laser melting: Materials and applications. Applied Physics Reviews, 2(4), 041101.

Yu, C., Qie, L., Jing, S. and Yan, Y., 2019. Personalized design of part orientation in additive manufacturing. Rapid Prototyping Journal, 25(10), 1647−1660.

Zadeh, L.A., 1965. Fuzzy Sets. Information and Control, 8(3), 338−353.

Zhang, Y. and Bernard, A., 2014. An integrated decision-making model for multi-attributes decision-making (MADM) problems in additive manufacturing process planning. Rapid Prototyping Journal, 20(5), 377−389.

Zhang, J. and Li, Y., 2013. A unit sphere discretization and search approach to optimize building direction with minimized volumetric error for rapid prototyping. The International Journal of Advanced Manufacturing Technology, 67(1−4), 733−743.

Zhang, X., Le, X., Panotopoulou, A., Whiting, E. and Wang, C.C.L., 2015. Perceptual models of preference in 3D printing direction. ACM Transactions on Graphics, 34(6), 1−12.

Zhang, Y., Bernard, A., Gupta, R.K. and Harik, R., 2016. Feature based building orientation optimization for additive manufacturing. Rapid Prototyping Journal, 22(2), 358−376.

Zhang, Y., Bernard, A., Harik, R. and Karunakaran, K.P., 2017. Build orientation optimization for multi-part production in additive manufacturing. Journal of Intelligent Manufacturing, 28(6), 1393−1407.

Zhang, Y., Harik, R., Fadel, G. and Bernard, A., 2019. A statistical method for build orientation determination in additive manufacturing. Rapid Prototyping Journal, 25(1), 187−207.

Zhao, D. and Guo, W., 2020. Shape and Performance Controlled Advanced Design for Additive Manufacturing: A Review of Slicing and Path Planning. Journal of Manufacturing Science and Engineering, 142(1), 010801.

Zhou, Y. and Ning, F., 2020. Build Orientation Effect on Geometric Performance of Curved-Surface 316L Stainless Steel Parts Fabricated by Selective Laser Melting. Journal of Manufacturing Science and Engineering, 142(12), 121002.

Zhu, L., Feng, R., Xi, J., Li, P. and Wei, X., 2020. A Lightweight Design of Tree-shaped Support Structures for SLM Additive Manufacturing. Computer-Aided Design and Applications, 17(4), 716−726.